MARRAKECH - Universal Acceptance Steering Group Workshop (UASG)
Sunday, March 06, 2016 – 08:30 to 17:00 WET
ICANN55 | Marrakech, Morocco

DON HALLANDER: So it's now 8:35 on Sunday morning, and we'll go ahead and start. My name is Don Hallander. And I'll just go through the program as it stands now, subject to change, depending on how people's schedule works out. So apologies from Ram Mohan who will try to attend as he can, but he was quadruple or seven-tupled [sic] booked because of his various roles in different organizations.

So the plan, first up we have an update on the UASG reorganization and achievements that we've done. And I'd like to go through the proposed changes to the charter, which have been distributed on the UA discuss list already. Then we'll have a report from [Cormell?] on their experiences. And then we're going to try for some live demonstrations.

Marvin has got a Windows based EAI client that he wants to show us. And we're going to try to send some emails to some EAI addresses from your favorite email client, whether it's browser or desktop or whatever, and see what happens. Then Ashwin will join us for an update on ICANN's progress towards universal acceptance.

We'll take a break. Rich is going to go in some depth into the technical work that he and Dennis have done in terms of the registry/registrar model. Then we'll have some UA initiatives from around the world. Dusan will talk about Eastern Europe, Raed will talk about what's happening in Saudi Arabia, and Lars is going to talk about what's happening in Western Europe.

Then we'll go through the quick docs, quick guides. And I've got copies here of a no longer current version but pretty close. And Mark will go through those. And then we'll talk about next steps and what we've learned from there.

So any questions, or anybody want to change…? We may end up changing the schedule a little bit. Ashwin may have to come in earlier or later, and we'll just… It's a small group, so hopefully we can just go slowly and comfortably. And with respect to remote participation, Wendy is going to let us know if there is any issues. Welcome Ram. We'll pass you the clicker.

RAM MOHAN: Welcome and my apologies for coming in a few minutes behind. Good to get this going. I think the first thing that we really wanted to do when we looked at… Hi, good morning. When we looked at UASG was to brief the community, brief all of you, on some of the things that we've done, and I think also to give you a sense of how we've evolved from where we started about a year

ago to where we are now. And to ask for your support and for your assistance in progressing from where we are.

I think we have some clear goals set up. We helped set that up last year, but the way we looked at beginning to do the work last year, to where how we're doing it now, is quite a shift. If you recall, last year when we, you know, when this kind of initiative came together, it sprang up from so many of us coming together and saying, this is an important thing. Certainly the introduction of many, many, many new gTLDs brought further urgency and relevance to the problem of universal acceptance. We know the problem has existed for a long time, but the relevance of it became in sharp relief when we had from, you know, going from a few hundred TLDs in the zone file, to you know, more than a thousand TLDs in the zone file.

But many of them that suddenly stopped working right, or right is a subject term, right? But working as a way many people expect them to work, in various applications in various systems, right? And USG got formed from there, sprang up from the community, still very much a community driven initiative, and one of the, I think, unique things about UASG is it's not an ICAN initiative. It's not something that ICANN gave birth to, or there was no Board resolution, or something like that that started up UASG.

It was not something that came out of, you know, the ccNSO or the GNSO coming together and saying, you know, let's do a policy development. It came from the community. And it's drawn from a wide cross-section. ICANN is kind of a natural place to host meetings, etc. And ICANN is also recognized that providing funding for UASG is the right thing to do.

But down the road, I expect that the funding sources will expand to more than ICANN, and that's a good thing. The other piece of it is that UASG's participants and the active participants have already gone quite widely beyond the usual suspects from, who just come to ICANN meetings. You know, we have organizations, companies, not for profits, academia, from outside of the normal spectrum of ICANN.

And I think part of our goal is to keep it that way and to ensure that the work that we do remains relevant in such a way that it attracts the attention and the energy of these organizations from elsewhere.

The third thing that we sort of doing last year was we came up with this idea that we should self-organize ourselves, and we went about it in a very workman like way, right? We said, let's take the, an engineering set of problems. Let's look at, you know, email address internationalization as a significant problem that needs attention. Let's look at the fact that there

are libraries and systems, and code, an entire code repositories that are making incorrect assumptions about what is a valid TLD or what is a valid email address.

Let's go and correct those. There are some operating systems and some entire programming frameworks that have libraries pre-built that need a little bit of update, so let's go focus on that. Let's start to do an outreach effort. So we began originally with kind of technical track if you will and then a marketing track, etc. with coordination groups, things like that.

And I think our realization, relatively quickly, was even though UA discuss has, I don't know, 100 people, more than 100 people, something like that. The core group of folks who showed up regularly on the email lists, who contributed, who participated, and who actually did the work, intended to be the same 15, 20 or so individuals in this first year.

And what we were doing was we were taking these 15 or 20 individuals and we were putting, you know, three in the marketing piece, and four in the EAI piece, and another three in the software libraries and the technology upgrade pieces. And to some extent, we were spreading our somewhat or our resources… The most valuable resource which is the intellectual capacity and the brain power of our people inside of the USG was spreading that out.

So we decided earlier this year, we met in January this year, and we decided that the right thing to do for where USG is now, is to consolidate our efforts and to not have four or five different sub-teams, but to bring our efforts together. That was one important [inaudible]. The second thing that we also did was we changed the decision making, not the decision making actually, the implementation of decisions structure, and we've invested a lot more of our energy and time in arriving at good decisions, and then tasking Don to go execute on those decisions, and then making sure that those decisions continue to go forward, right?

So I think, as a result, you'll find this year that you will find the gap between ideas and execution to be far smaller, to be a much shorter gap, simply because we don't have that whole process of, we've got the idea, let's get the consensus of the idea, now let's go look for the volunteers who can help make the idea happen, and let's wrangle schedules.

So Don has moved from relatively speaking a more administrative function that he was doing last year, to a far more execution function. And it has led to a very, much more happier Don as well, which is a good goal.

I'll go through the slides, just I don't think I necessarily have to go through what is a definition itself, but this is something that we have arrived at. We came up with, it has been sent to the

discuss list, so this is recap for you. We have now a definition of universal acceptance. One of them, and I'm not going to read out what's on the screen. I think the important thing to point out is that last year, there was some confusion at some times that universal acceptance was the same as universal awareness.

And there was this assertion being made that universal acceptance, one of the things this effort ought to be doing, was to also spread awareness of new TLDs and the applications and things like that. And this is a bit more constrained, if you will, compared to a universal awareness goal. We don't think we're the right group to be working primarily on universal awareness.

I mean, obviously, as a part of universal acceptance, there will be awareness raising, and in fact, if you've looked at the budgets, if you've looked at the, of what we put out, there is, especially this year there is going to be quite a significant amount of awareness raising. But the raising of the awareness is on acceptance rather than just awareness for the sake of awareness, okay?

Questions on this? Any comments on that? Okay.

RICHARD MERDINGER: Rich Merdinger, one of the vice-chairs of the UASG with Go Daddy. And what I would ask is if, if we're going to be focusing

on the more constrained definition and how awareness applies in the acceptance realm, meaning awareness of acceptance, do we have…? I should be able to answer this question but I can't.

Do we have a partner that is going to focus on the awareness of the TLD situation in general when it comes to new TLDs, IDNs, EAI, etc.? The more commercial side of it.

RAM MOHAN: I don't think we've gone through that discussion yet. My sense is that, we'll be doing some of that almost as a default because… I mean, I'm drawing, I'm pointing out a line, if you will, but at best, it's kind of a dotted line. It's not some hard line between awareness and acceptance. But the primary reason why I am separating out awareness from acceptance is that the awareness job is a huge job, and it's relatively easy, I think, for the awareness job to overwhelm the acceptance job.

And I think success for us would be measured much more by ensuring acceptance. And if we have acceptance, then I think it's actually a far easier sell to say you should be aware of this, these names work. These names work in the browser, emails will work, etc. mark?

MARK: Mark from Microsoft. I'd like to reflect the question back on Rich a little bit. People may know that Microsoft and Go Daddy are actually partners in Office 365. Go Daddy is a Office 365 syndicator, meaning that you can hire them to give you a domain name, assign that as your Office 365 domain name for your company. Set up all of the email addresses like that, and then they provide additional services as well. And customers are very happy about that, and it occurs to me that when we do have a certain amount of universal acceptance, that Microsoft and Go Daddy would be partnering to make customers aware that these names are available for use within an Office 365 situation.

So I think they will just organically be opportunities like that that come up once we know that the systems are stable and available.

RICHARD MERDINGER: Yes. So from our perspective, Rich, we're definitely not an awareness raising with a broad consumer base, but we do want to raise awareness with the issue of our target audience, which are geeks, software geeks, more than network geeks. So that's our target audience. Once we achieve that, I think it will be much easier for the people who are keen on universal

awareness, which will include registrars and registries to go out and promote their products.

But right now, I'm hearing, people are a little reluctant because they don't work as expected. So my goal is to get this fixed sooner rather than later from a technical perspective, so that the registries and registrars can go out and have fun. Does that make sense?

MARK: Yeah. Very quickly, I would say that the reason I brought it up entirely is because of the complementary nature between awareness and acceptance, and the fact that having the acceptance and even the work products that are being developed by the universal acceptance group can be in the hands of the people that are trying to do awareness, and this does need to be done first.

And so I think the approach is definitely right. I just, it felt like we were saying we're narrowing it a little bit, which is true and appropriate, but we're not actually constraining the whole problem space. We're just focusing on part of it first. And we may not be the ones to focus on the outer concentric circle, but somebody will need to do it and we're going to provide them with some ammunition and food. So that's good.

RAM MOHAN: Yeah Rich, I think that's exactly right, and I imagine that we'll be partnering with multiple organizations and multiple initiatives in different spheres that are affected by this. So I would imagine, just off the top of my head, I would imagine, for instance, that we, you know, we would go and work with folks in the anti-spam area to ensure that there is acceptance and there is knowledge of what these strings mean, and what the patterns there are, because today there are some strings that kind of automatically make it on to certain filters, because of rules.

Not necessarily because they are performing in a certain way that have them deserving to be on list like that. And that's awareness in a different manner, but I expect, you're right. Those outer circles were, I don't think are goal is to be focused on the outer circles. Our goal is to get the core off this thing fixed, or worked on in a significant way.

And there will be awareness as a part of what we're doing, but as Don said, the awareness is more targeted awareness.

So we've… As a group that is focused on building on a core of materials, reference, information, things that are usable, reusable, citable, documents that engineers can use. Documents that managers can look at. Documents that executives can be pitched to. I mean, those are the broad scope

of things of the set of documentations that we're working on. And up there on the screen, you'll see we are up to UASG 08, and which is in final review, seven is currently, this afternoon will have another detailed reading of it.

And it's a fascinating exercise. For those of you who have not gone through detailed readings. It begins sounding like a tedious thing, but you actually realize once you engage and you get into it, that the words actually have tremendous meanings, and getting the words right makes a big difference. And the interactions end up simplifying the intent. And that's a big job.

So there is, these have been published. They will continue to do that, and we're also starting to look at which of these ones need to get into translations and things like that. We're not taking an approach that everything we write must be in every language, etc. I think that's… When…

I think our goal is when there is demand for it, we will be able to go and have these documents be available in the languages that, where the demand exists.

What you see there are the things that are in the work plan for the next few months. I don't even think this is the all of 2016 work plan. There are more items on the 2016 work plan for UASG. But the main activities that are going on right now. There is a review that is commenced of the most popular websites for

UA readiness. There is a review as well as remediation work that is beginning on the most popular development tool sets.

Also looking to build out both use cases as well as test environments, and what's written there, building the UA community, to some extent, there is already a community, but the job is to help bring coordination to connect the various parts of the community.

And those we are doing EAI in say the Arabic script, versus those who are doing EAI in the Thai script, and those who are doing EAI in say the Han script. To be able to pull all of them together and to share common experiences, and to leverage that to build some level of, not just inter-operate-ability, but understanding the nuisances that exist from one to the other is an important thing.

And I'll note that there are some… The EAI piece is quite a layered piece, because the standard, so to speak, underneath themselves have some interesting, there are some interesting evolution that still exists, or still needs to happen. There are some Unicode, interesting things happening with new versions of Unicode coming out, and the prior Punycode standards and how they interact, things like that.

And there is also a planned review of browsers and operating systems, and a white paper. Don, do you want to take a minute

to speak about the white paper and what the goal of the white paper is?

DON HALLANDER: Sure. Thanks Ram. So the white paper has been in various people's visions of different things. And I think we're coming to a general agreement. It's a document, first of all, to provide introductions, but also for people to read and say, there is a real valid reason, commercial, cultural, political, to pursue universal acceptance. And we're using a couple of existing documents as guide books or models, and greasing the wheel work that ICANN did a couple of years ago, the [inaudible] annual report on IDNs, the ISOC, I call it the ISOC blue book.

I'm sure it has a proper name, but I call it, but they produced a couple of years ago with stories about how this works. So that has been circulated to the UA discuss list, looking for feedback. And after Marrakech, we will put it out for, through our help wanted process, RFP process.

And we expect to get some responses and get it underway, I think, by the end of April, with a goal with finished this financial year. So we don't think it's going to be a big, huge document as some of those others are, but build on that work and that approach.

RAM MOHAN: And the idea behind the white paper is to have something that is, that has gravitas, if you will. And in a couple of ways. Not just in the subject matter and the content in there, but also who is writing it. Making sure that you can go out and say, this is a paper commissioned by, and we're just making up names. We haven't selected who, but imagine if it's a paper written by [Godner?], or a paper written from the Media Lab, or something like that, where the organization already comes in with a certain level of reputation that attaches to the white paper.

The intent would also be that when we engage with people who are somewhat non-technical, but who have a decision making role, such a document would be invaluable because it's not something they may even read, but it's something they look at and say, oh, this is from Harvard. This must be legitimate, and then they pass it on to the person who is actually going to do the work. So in many ways, it's intended to open the doors and let us in, and build that credibility, because UASG itself is, I mean, nobody knows who it is, and trying to get an audience with somebody, even mid-level if not a senior level At-Large corporations is a significant challenge.

Administratively, Don mentioned a little bit of that. I'm not going to go through a great deal of detail here, but there is, you

know again, small steps, but steady steps and steady progress that we're making on the administrative side. All of these things get documented on email and so if you follow along, you will see these things come through.

The budget process has been non-controversial. One of the things that in the 2015 timeframe we were former ambitious, we put together a pretty large budget, but what we realized pretty quickly was there is no point in having money in a budget. You can't really spend the money if you don't have stuff to go and talk to people about, right?

So that's why a lot of the work has shifted doing the work, having the documents, you know, getting the basic information. Because once you have that, then you can go to the conferences, then you can go to various places, then you can commission a paper, things like that, and actually move it forward. And we don't operate on a model of, the money is not spent, it's gone. That's really not how it works.

If the money is not spent, there is still an opportunity to bring back money for the next time around. So I think I've already gone through this. This was a significant shift. And one other thing that I wanted to mention was last year, there was some discussion about UASG regional, if you will. So to spark off a UASG in China, and a UASG in India, and things like that. We've

had some discussion, and I'd love to get your input on this, but the thought process at this point is that, I think, we think, UASG actually has to get the core job done and focus on that, focus all of its attention on that.

But what it really ought to do is to encourage and provide support, by providing supporting materials, things like that, for regional and national initiatives, rather than taking on the job of building out, you know, kind of regional franchise or a national… That's really not the core model. So the incredibly supportive of local efforts, regional efforts, and provide as much information and expertise as possible in there.

But not really actually make it, you know, UASG Asia and UASG Europe, things like that. That's kind of our, what we're thinking, would love to get some feedback from you before I move on to the next slides. What do folks here think?

I'm seeing some nodding heads, but anyone who thinks it's a wrong approach or there is a better approach than that? Edmun?

EDMUN CHUNG: Not putting my hand up to say that it's a wrong approach. I think, you know, having local and regional groups to really go out to, you know, on the ground to advocate UA is very

important. And I don't think there needs to be any kind of very formal delegation in that manner, and therefore no need to monopolize saying this is the UASG Asia or China or anything.

But we should definitely encourage multiple initiatives in fact to reach out because, in fact, one entity probably will not be enough even for, especially for places like China or India, which is in fact itself is a continent. So that I think is important. I don't think there is need to accredit a particular regional or local, but we should definitely encourage multiple of that to happen because we need them on the ground to actually do the advocacy.

RAM MOHAN: Thank you. Other comments? Don.

DON HALLANDER: Just speaking to Edmund's point. One of our documents is a suggested starting guide book for regional, and local, and national UA initiatives, and in next year's budget we do have some money to help seed some of those, it's certainly not enough money to operate one, but if it, you know, if there is a group that wants to get going and they need a bit of seed money to get started, and we do have that in the budget, but we don't want them to be UASG, China, U.S. UASG Italy, or UASG Soma.

But we do hope that there will be good communication back to the global group so that if UASG Soma does something, then the people in Italy who are also pursing UA can say, "Oh, yup." So that's something that they can learn from, but not in a franchise model at all.

RAM MOHAN: It may surprise you, but at some point we may hope to wrap up UASG. You know, this is not something that we want to perpetuate for, we want to get this… Success will come by, you know, by having this group wrap up, because you know, it's actually out there, it's all mainstream.

So on the just, I'm going to move relatively quickly through the rest of the slides here. The slides themselves have been published, and I want to just give you the highlights. On the charter itself, as I said, the project group model that we are working on, we've moved that idea because it's…

We may yet come to that idea, as things scale down the road, but we figure if that happens, we can go and add that back in. But we didn't need to have something in the charter that then constrained us to work in what was becoming a relatively inefficient way. And the other thing was a supportive group of volunteers to support paying group of volunteers.

Although it's a small change, it has a big impact in what it really means underneath that. Don, I'm going to pass the clicker back to you. I think the clicker goes to Ashwin next.

DON HOLLANER: If that's alright.

RAM MOHAN: Absolutely. I mean, yeah, Ash has something other commitments, so I just wanted to say thank you for coming Ashwin. One of the things that we've asked ICANN to do, and Ashwin has been gracious enough to do it with us, is to share what the universal acceptance story, if you will, at ICANN, and what kind of efforts and what kind of challenges that he has been facing.

So with that, let's get it over to you Ashwin. Ash is the CIIO of ICANN.

ASHWIN RANGAN: Thank you Ram. Ram insisted that I wear a red tie, that's the only reason I'm wearing one today. Sorry, both he and I have an appointment to meet with a high level dignitary that's visiting ICANN today, and I've been given the responsibility of receiving

him in about less than an hour's time. So I appreciate the fact that you have been flexible with the schedule.

And Ram thank you very much for accommodating my schedule requirements at a late date. I appreciate it very much. Thank you. You've been hearing me talk about what we have done from a UA perspective at ICANN for the last two, perhaps three meetings. It's a laborious effort. And it's a work of love.

We committed to this. We want to see it through to its logical end for very obvious reasons. This is the way to bring the Internet to the world for us in many ways. We've continued to make progress, and what I show you here is going back a little bit, and then going forward from there.

So the four or five topics that I want to cover today are, what is UA again for those that may be in the room for the very first time, and why you should care about it. I've talked in the past about how we have differentiated between our custom built services, and packaged services. Where we have source code control versus where we don't have source code control. And the approach to getting UA compliance is quite different, depending on which pathway you've used to procure software or to create software.

And then a quick status on UA at ICANN, and a glossary of where you can learn more. These are all of the things that we have

been leveraging as we've gone down the path of learning more about how to become UA compliant ourselves. The definition of UA, I think, was frozen in the last year or so where the exact words were fixed. We were looking at a few words and examining whether or not they belonged.

I think now what we have up there on the screen is the definition of UA, where it is a state really more than anything else. It's a state where all valid domain names and email addresses are accepted, validated, stored, processed and displayed.

I think those are the five verbs that define whether or not you become completely UA compliant. And it was important for us to fix those because that then tells us when we are indeed successful at making something UA compliant, or determining whether or not something is UA compliant.

Now there is the correctness of it and then the consistency of it. The correctness is really within each of our own four walls. The consistency is when it crosses our wall to somebody else's boundary. And it's there that we all need to come together. So it's one thing for me to say I'm UA compliant, but if my email doesn't go past my boundary because somebody else's boundary stops at their boundary, then we are not UA compliant.

And I think that's the big challenge that we all have together. What's different now, I think many of the folks in the room have lived through the evolution here, where in 2010 the IDN ccTLDs came into being, if you will, wave of new gTLDs that said that the domains to the right of the dot are beginning to become more and more complex. And then the internationalization of character sets.

We started more and more of those. If any of the web applications or software does any of the following, then UA becomes important. So if the web address is URLs accept or store web address, the site address if you will, ought to link to websites or validate websites, URL in forms that are submitted for workflow processing or file transfers via URLs, or email addresses that need to be validated, send and receive obviously. Login with email.

We were having a story session, and one of the stories that was built up in the course of the discussion was a very interesting one, where someone said now, if an Arabic speaking native were to land in Heathrow Airport and wants to log into his language website from an English browser, can he do that?

And we said, "Wow." You know, that's an excellent example to capture, because that just tests every piece of the infrastructure in the system for UA compliance. So it was a wonderful story

that we captured immediately because I was like, wow, the lights went on now. So to get there is really hard. It's really hard.

These are just a few examples. Top level domains to the right of the dot, and secondary domains to the left of the dot. The length characteristics are now pretty well understood. Entire domain definitions at the URL level, that is also pretty well understand so that we can check a box and say, if it has this, then we know it's UA compliant.

The character sets of course, will continue to evolve as more and more characters and more and more languages are clearly seated in panels, and the language scripts and definitions are finalized. With custom code, the user interface is something that becomes really important, whether it be for capture of data or display of data. Handling the number of characters, we're seeing some of the packages that we use for developing custom code, they themselves may have limitations.

They may say that this input character field can only be X wide. So we're having to go through that exercise of determining which ones are easier in that regard and which ones are not. Now there has been quite a bit of discussion in this room about Unicode and Punycode, and we've taken a lot of that as input

from this group, and we now are in a position to treat our own software in particular ways, so in validating TLDs.

We're saying the short answer is don't do a look up table, because what we now have as just the first wave of gTLDs, it's only a matter of time before there are others. So if the lookup table is not updated, then you may be rejecting something that's in fact a valid input text or field, but validate instead using Unicode or Punycode.

If almost all of your business in ASCII, then yes, say in ASCII, convert to Punycode is needed. If not, then convert everything internally to Unicode. I think that's a fairly clear definition that we've arrived at. Now that works for us, because most of our services have been written with English as the de facto, ASCII as the default. Now that may or may not work depending on how you start building your own software in the regions that you come from.

At a system level, it's more about search, process, and export. It's not so much about accept, validate, store, and display. So it's a different set of parameters that we're using from a system level. Search becomes really important because the way in which search works, depends on how the storage has occurred in the first place, and how the interface between the search engine and the stored data in the database interoperate.

It's a tricky area. It's one of the trickiest areas that we are seeing. You know, the accepting storing becomes relatively straightforward, but when you get into searches where you start stumbling across… I didn't mean for that to happen. So we have to think through the use cases of what did you actually mean to have happen? And therefore, figure out how best to store it, and how the search engine will accept whatever is being entered, and how it will go across the database and search for things and bring it back.

For program to program communication, we've decided to convert everything to ASCII. This way there is a common denominator, and this will be the way that we process anything that comes in program to program.

The other side is off the shelf software that we've either have licensed for use, or we have purchased and we pay an annuity for use. So one would be [sass?] the other would be commercially off the shelf of available software. If they have domains and emails themselves, we're asking our supplies to become UA complaint. Many of them do not know what UA means.

We're having conversations with them. The last time I was before this team of people, I said that that is a double edged word. On the one side, we can educate them about UA, but the

moment we talk about it, it opens the door for new negotiations potentially, from their perspective. So it's a very important, but very tricky though to get past.

In some cases, they understand why UA could be, from a business viewpoint, a high potential offering to their broad base of customers. In others, they look at us and say, will you fund our upgrade? And it's like, you know, it's not just me. It's your whole customer base, so that becomes a tricky conversation.

Where we're pushing them is to commit to a deadline. We're saying, if you don't have it, it's fine, but tell us when you will have it, and tell us what it takes to get there. I think we just lost connection, Don.

ANDREW SULLIVAN: While we're working on technical, do you want to question?

ASHWIN RANGAN: Please.

ANDREW SULLIVAN: Okay. So there are two things that I wanted to, I'm Andrew Sullivan. There are two things that I wanted to just catch on this. One is, if you're giving this presentation to other people, you have several places where you say like 255 characters or 63

characters. Be very careful with your audience, right? Because some of those people are going to hear characters, and they're going to think like, for instance, you know what… And they're multi-octets, right?

So the limitation is a hard limitation in the DNS. And the 63 or 255, but that's ASCII. And not everybody understands that the Unicode presentation, the U-label is translated into this underlying thing. And so you just want to be careful with that because people get deeply confused about it, and then you have a long, boring conversation about how we did it wrong.

And the answer is yes, but too bad. The other thing I would say is that, ICANN is in a special place, you say that you've decided to put everything in ASCII, and it's important because you're dealing mostly with domain names, so you have that choice. But if your primary problem is storing emails, you won't have that choice because there is no ASCII compatible mapping for the local part.

And so it would be great if you were giving this presentation to others, to highlight that thing, that you really need to analyze your use case, or people will, you know, you'll commit yourself to a direction that then you can't back out of. So that's… Those are just two comments that I noticed on the way by.

ASHWIN RANGAN: Thank you Andrew, thank you. I appreciate that. You know, this is a case of you sort of lean back on the comfort that you have. [LAUGHTER]

Thank you.

Thank you.

We've also looked at what else we can do when we open the contract door, and we're taking advantage of the fact that since it's open, we're asking our suppliers to become IPv6 compliant, DNSSEC compliant, and you know, there are very interesting conversations when we go through this.

I'll give you an example case of an IPv6 compliance conversation. We recently deployed some software that's IPv6 compliant. Then we made them aware of the fact that it's a [Sass] product, and we made them aware of the fact that we're a very global cooperation, and we need edge accelerators for content.

The CDN that they leveraged is not IPv6 compliant. I tell you, I mean, this is like an engineering nightmare at the infrastructure level because you're talking with three different sets of engineers, who choose not to speak to each other using the same terminology. So they all three go away confused, multiple

times before they can come back and decide on what is the problem and who has the problem.

It was frustrating and interesting to watch this human interaction at play. I mean, the technology is very clear. The definitions are very clear, but people would get in the room and they couldn't get to who has the baton at any given point in time. So, multiple iterations later, the party that really had the work to do on their part, said, okay, we will take this up and it will take three months, and we will work with the CDN and we will make it happen.

So it's really interesting. It's yet another area where there is potential for confusion.

Okay. So this is a set of example contract paragraphs from one of our contracts. I thought it may be helpful to see how we have written up, thank you. To see how we've used contractual language to approach our suppliers. Now, I know that this cannot become boiler plate. It depends on where you are in the evolution, but I thought it may be helpful to share our example in case you haven't started down this particular pathway.

This will at least give you a starting base. And you know, working with your particular legal teams, of course, you would have the opportunity to modify and move forward with this.

Did that repeat? I think it did.

For us, I've shared that we have 84 services that we have been slicing and dicing in different ways. Our off the shelf is 40 of these services. 21 of them need contract updates, and those are the ones where the door is slowly being opened up with our suppliers. 18 of them, it doesn't really matter because they don't handle URLs or emails.

But we had to go through extensive analysis and make sure that they did not have either URLs or email connotations. And one of them we found was compliant, and I'll talk about how we went about with the other side of the house which is custom. 32 of them need to be tested, and they will need to get fixed over a period of time.

Seven of them, there is nothing to be done, again, because there is nothing that is UA oriented in them. And five of them are UA compliant, to a large degree. Not completely, but to a large degree. So the example of the five that are in compliance is on the back of ourselves, first platform. We leverage that to create and deploy several of our services.

Some of the things that we need from a UA perspective come out of the box with them, as we do what they call as declarative programming, which is another phrase for configuration, UA compliance comes out of the box. And some of the limitations

that we finding are in field lengths or label lengths, which is interesting.

I mean, you wouldn't expect that to be the case. I mean, it's a database you declare and it's whatever it is, but now they have global limitations, which override anything that's locally declared. And the global, when it overrides the local, suddenly truncates, and if it truncates either from the right or the left, then their language connotations depending on how the truncation occurs.

So we fed that back to them, and their committing to delivery with their next releases. Now, the confidence that we have in their delivery will increase as we see the first delivery happen that says we fixed this. You know, it's that that we're going through there.

These are all of the resources that we have been using over the last three or four months to become better convergent ourselves, and to make our teams better educated about UA. We are taking our contractors who do most of the programming for us through these lessons, so that that's becoming part of their muscle memory. On the back of a request from Dawn about six months back, on the back of the work that we're doing too internally, he asked that we create a CIO's guide to becoming UA compliant.

And we're diligently working towards that. A couple of chapters have already gotten written. I suspect that by the mid-2016 timeframe, it will become better. We might even bring it out towards the end of the year, so that this team can take a look at it and provide feedback, so that it becomes a useful resource for all of us.

Our rough timeline, when first I spoke here, we were still taking inventory. We've made quite considerable progress since then. What we're finding in working with our business units is that, when we recognize that something is not UA compliant, and we have a conversation with them about driving to UA compliance, if there is a project on the table that touches one or more services, adding stories about UA compliance, is easy.

But UA compliance is not the reason for which they would like us to touch their service. So that's a very difficult conversation to have. Because the perceived value of becoming UA compliant is not necessarily a story that they are willing to buy into, for fear that something that's working might actually get broken in the process.

So that's a very real sort of human reaction that we're getting when we have these conversations internally. That has led us to back away from a commitment that I've made here saying that by X, we will be UA compliant. Instead now, I'm willing to say

that we will continue to do this work as strategy and time permits, because I think the time permission is something that we negotiate with our business units, so as their strategy for either re-platforming [sic] or re-rights come up, and as time is given to us, we will continuously do this work.

So this may take longer than we had previously committed to here, but we will learn our lessons and we will feed that back here. As far as new software being always UA compliant, on the in-house side that is a commitment that we have made to ourselves. We're saying anything new we build, will be UA compliant by design.

But as far as new software that we procure off the shelf, or we use as [sass], we don't know. We have the same issue that I told you a story about where we bought into a [sass] service, they were UA compliant, but the CDN is not. We're finding that CDN is, by the way, a common denominator, the [sass] where [sass] is UA compliant but the CDNs are not.

We've seen that in multiple cases, multiple different CND providers. So that maybe a high leverage point for driving the UA compliance overall.

That's our story, so far.

UNKNOWN SPEAKER: Questions for Ashwin?

EDMUND CHUNG: Thank you again for the very comprehensive update. It's always fascinating to understand that, you know, step by step really, at least every three or four months we get an update. That's great. And your point about, you know, the first time you said, I think it was 18 or 20 something months, I was in my mind saying, I don't think they are going to make that. Let's see. And I think, and the particular point is procuring other software in-house, new software that's probably a reasonable target. And that's very, very interesting.

But, you know, those are a couple of observations. I do have a question, and actually building on what Andrew asked earlier. I see in one of the parts you said for a machine to machine, you're kind of using ASCII. That is the mode. I'm curious about the email side of things, when you pass email addresses along.

Because that doesn't, is not possible to translate to ASCII readily. There are, of course, ways of maybe UTF-8 in escaped format, but that makes it interesting. So you know, that might be something that I would like to know. And on the, you listed some clauses on, for suppliers. I think that's very interesting.

I'd like to take a look, because I will look at the… A very quick glance at the UA component seems to talk about email address, and then the latter portion omitted the email address and focused only on the domain names. I am not sure, was that intentional, or there were, you know, how is that crafted?

So two questions. One on the ASCII and one on the domains and email.

ASHWIN RANGAN: Thank you. On the first question, it's not machine to machine, it's program to program. So with program to program, we deliberately chose that. So if it's inter-process communication, or program to program communication, we chose to go to ASCII internally, between our systems and services. So it's not email, it's across our services when we pass either a URL from one service to another.

So it's within the four walls of ICANN so to speak. I hope that that clarifies it.

EDMUND CHUNG: I meant the email address being passed back and forth, not just sending email out.

ASHWIN RANGAN: Got it. Yeah, even internally, we're able to do that only because we're doing this… So we had those, if it is this, then go this way otherwise go that way. So I was trying to describe sort of decision three words, and I don't think I came across well enough there. So it's the Unicode, Punycode, we go this way, we go that way. That's the sort of combination that we're going through in order to pass these along.

One of the things that we're seeing is that the number of services where we store emails is relatively small, and the number of services that we pass along email IDs, is even smaller. So it's really specific to the cases that you have in the size of the portfolio that you're dealing with. If you want the specifics of how we're doing it, maybe the next time I come around, I'll bring an example of that itself. I will do that.

I think that may be helpful for this discussion. On the contractual clauses, as I said, this is one specific contract that we have. It's not a boiler plate. It's one specific contract that we have that I just abstracted the words from, and brought here. We are finding that in no two contracts are we able to use the same exact terminology, because this is a negotiating term right now.

So we're having to negotiate words and phrases with lawyers on both sides, that's the state of play right now. So it is not boiler

plate. I wish we could have a boiler plate, that would be wonderful. Then it's says that we are all becoming UA, sort of universally accepting UA.

That's a serious regressive kind of comment. [LAUGHTER] UA.

DON HALLANDER: Yes, so I thought your observation just at the end of your presentation, which was your internal clients were not terribly keen to change because they didn't see anything broken. As I talk the CIO around the place, there are saying something very similar, that they get it, they understand it, there is no pressing need, that they're going to put it on their list of things to address any time they open the hood, but they're not going to open the hood just for this unless they have a problem.

So that's, I think, why we're working to get these quick guides out, working to get the technical documents out. So that when you do open up a system for maintenance or a series of minor enhancements, that you just add this to the queue. In my experience, when I was doing a similar job is, you always had things to do when you opened it up, but was it worth opening it up just for that?

So I think you found reality.

RAM MOHAN: Yup, thank you.

UNKNOWN SPEAKER: Other questions?

UNKNOWN SPEAKER: This is [inaudible] from Pakistan. I'm representing taskforce on Arabic IDN. Is it possible to distribute awareness part into business and to the community? Because if a demand from the community will come, then business will automatically drive up their program, and they will ask for the fund for the [advancement?].

ASHWIN RANGAN: Yeah, I think that's a very relevant comment, and it's exactly in line with the mission and the goals of the UASG. So absolutely happy to work on that and to not only develop the materials, but to make sure that identified communities get access to that material. And as I mentioned earlier, including in local languages if that helps.

So we'll end up doing that translation work if necessary. If it assists in accessibility then we must commit to doing it, then we will commit to doing it.

KURT: So I wonder if part of our outreach should include the legal profession. You know, they have continuing education requirements and start to raise awareness about, these are possible terms you can include in your software purchasing agreements to ensure your software everywhere. And that would start the conversation going in another profession about, you know, what the heck is universal acceptance and what it means.

When they're talking about contractual terms, that might, you know, that's in a language they would understand and raise awareness there and get them talking about it.

ASHWIN RANGAN: Makes sense to me. Folks have any comments on that, on Kurt's suggestion?

Andrew?

ANDREW SULLIVAN: So I think, this is Andrew Sullivan. I think the key point here is exactly this point about [sass]. That the more stuff moves into the cloud, the more dependent you are on like this entire chain of all kinds of vendors. And if what you think you're writing is an agreement with somebody, you're wrong, because they're using

on these other people as well. And so you have to understand the entire chain that you're dealing with.

And if you're trying to manage a contract, that's a nightmare. Because you know, it basically you're suddenly in the position of having to micromanage the subcontracts of other people. So I think that this is, this is a point, I think, for lawyers when they're writing this kind of stuff.

They need to understand like what the consequences are downstream. And I, just from where I sit in the stack, I don't have the impression that the legal industry is completely up to speed on how the environment is working, so I think that can be useful.

UNKNOWN SPEAKER: Thank you. Don.

DON HALLANDER: Don Hallander here. I think that's something we can work with. What Leslie [Dego] calls the broccoli group, which are things like IPv6, DNSSEC, these are things that are good for you that you don't necessarily want, and we might be able… I know ICANN has an IPv6 clause, and ICANN, you may have a DNSSEC clause, I don't recall.

My domain names are DNSSEC signed. But there are these things, and it could be that we can work with that vegetable group to get some standard clauses.

UNKNOWN SPEAKER: Thanks. Edmund?

EDMUND CHUNG: So just building on those. Really, I think, again I like this topic in terms of, especially for those who would go out for tender and get suppliers, because it's a much better environment in a tender position rather than just, you know, negotiating the final contract. So you set out the tender in that way, then the respondents will need to tell you what they have.

So that government, and also, you know, larger corporations, and also government funded projects. There are plenty of government funded projects that require you to tender out pieces of subcontracts. So that's one thing.

Ashwin, you mentioned about the DNSSEC, v6 UA confusion. And it's very interesting to know, and I want to know more. We'll probably run out of time, but is it more effective to bring all three to the table, or is it less effective? Yes, it might create more confusion in the beginning, but it actually creates a bigger issue

so that the company understands that we're behind in a few things.

My experience talking to some companies, not on a, like supply situation but just in conferences generally, is that to bring all three to the table actually helps. Yes, it creates, internally, you know, it's like getting a big bomb into the company, but actually helps eventually get to the state where you mentioned that, okay, so these are the things that our team does.

This is your team, and you know, is it more effective to bring all three to the table or less effective?

ASHWIN RANGAN: We're finding that it's more effective to bring all three things to the table, because I think we are finding that to be leverage on our side. Your point is exactly right in that some of the [sass] providers, for instance, when they walk in, there is a swagger in their step. And they feel they have leverage on us as the buyer, but when we trot these things out and we start asking probing questions, they go back on their back foot.

And that's leverage on our side. So yes, definitely, it makes sense to bring all three out at the same time. Not one at a time, but all three out and say, we want you to be here, or show us a roadmap to get here, before we consider you on the possible

suppliers list. That really puts them on their back foot. So that's the short answer to that.

Your other point is very, very well taken. I think one of the highest leverage points is when governments bid out with suppliers for contracts. If these clauses were to become not required, but at least show us your roadmap clauses, that is a huge swaying factor I think, because that's a concentration point for many of these service providers.

UNKNOWN SPEAKER: All right, let's conclude. Thank you very much Ashwin, and I look further to updates from you, and I'll pass the baton to you, Don.

DON HALLANDER: So I heard.

So Ashwin, thank you very much. And always good to hear. So we'll go back to our originally scheduled programming. And we have Marvin Woo from Core Mail here, and he's going to talk about Core Mail's business and how they became the first commercial EAI provider of any size. And let me just get your slides up.

You can stay there.

MARVIN WOO: Okay. Hi.

Okay. Let's begin. Thank you Don. It's commerce presentation. I'm Marvin Woo from Core Mail, China. Now, my presentation is four parts, who is Core Mail? EAI and Core Mail, EAI commercialization progress, and challenges and plan on EAI.

Core Mail is not a famous company, so I need to [inaudible]… We are founded in 1999, the same year as IDN. We are twins. In China, we got 700 million users, and during 17 years, we working on email only.

This is Core Mail's management team. Maybe a little pageant, a little sunny. May who which one is me? [Inaudible] in picture. Okay.

Always, someone ask me why Core Mail tries to work on EAI so earlier. And, I think that this problem may be need to help more people to use email, because in China, some people can't understand English, such as my father. My father is an old farmer. I work on email, if more than 15 years, but my father can't use my program, never use my program.

So until I give him EAI account, Chinese email account, this email is the first email my father sent email to me after I work on email 15 years later. So I think EAI will help people who can't understand English to use email.

Okay, next one. Under the [inaudible] is in China, our company always in Internet, maybe is follow, but maybe some earlier just like EAI. I think we can become a leader. So I just choice to support the EAI first, and so earlier.

In 2012, Core Mail will, and their first email from the IDN email account that was sent by Professor [inaudible] and the least email was support by Core Mail.

In 2013, I will attend the workshop of APEC TEL48. And CCNIC in the same year. CCNIC 35 in the EAI commercial email system.

In 2014, CCNIC released the world's first Chinese IDN email account registration platform, and this platform was made by Core Mail.

In 2015, Core Mail promoting EAI at Wuzhen summit.

So how to implement EAI? We survive double account, double email address. Main is for IDN, and other areas is for English. Okay.

When receive email, the system will be judgment. If the email is IDN or not IDN, if can't, we are used two alias ASCII code. Okay.

So, the same is the same. If the system accepts uni code, we are used to RFC 6532, but if [inaudible] I will send aid for ASCII code.

Now, so in 2012, we can support EAI, but we have so many challenges. The first is always someone ask me, I like EAI, but no place can register EAI account. And then someone ask me, our system can't complete TP EAI. So no client can support, no client can support EAI. So many challenges, so we do something.

Okay. We release EAI register platform. So registrants can register some EAI account from our platform. And the release Core Mail's flash email, flash mail, APP Lunkr [inaudible] to supporting EAI. So our client include IOS, Windows, [inaudible] to support EAI.

And we upgrade email clouding service and platform to support EAI. So, Core Mail's [inaudible] platform, all of them can support EAI.

This is the platform. People who can register EAI account with Chinese. This is [Chinese], it's Chinese domain. It's EAI account. All of you can register, if you want.

So the [inaudible] is our client, and it's just use for IOS [inaudible] Windows. All of them can support EAI account.

Two is for mobile client, one is for PC client. Next. And also, we have some country to program promoting EAI, just like Thailand and India. So we support Thai languages, EAI account and Hindi,

India, so Hindi, Hindi language to use EAI account. This is a test mail.

Indeed they can work now. Okay.

And this is some commercialization progress with our EAI commercial. We can sell some [inaudible] for my custom that is who is registry for new g, new g registry. That the old EAIs planned from the software is used Core Mail. Our commerce [sass] platform, and is can support EAI account.

So maybe more than one million users can yield EAI account that with the Chinese. Also, we can catch some many ways, yeah, maybe [inaudible]…

Okay. Next. And during the three or four years I work on EAI, maybe I have so many challenges. And the first is few, always people use email. Someone is to send email and receive email, but so many people use email as registered ID. Just like this meeting. Our register ID is the email, but few system can accept Unicode like ICANN's meeting.

I just tried to register with my Chinese email address, but they can't accept. So most challenges for me is few system can accept Unicode. So I can't use either. And just like IPPs, some games, games, PC games, or mobile games, or online games,

they also use email register this ID. But the system always can't accept EAI.

So most challenging for me. Maybe Mark has the same trouble. And another compatibility of some software, just like EIP, OA, and CI. Just like this system. They also can't accept, they only accept ASCII code. So it's a fairly big challenge for us, because for the custom, the EAI is very important, email is not.

So there is always [inaudible] for ERP, not for email. So we have fairly trouble. Okay, the next one.

Okay. In the future, we have some plans is further programs via email in China. And I think maybe to promoting EAI will be in a closed cycle, such as government, they are always in a closed areas. Maybe they can accept EAI.

So the first our tries is the government or [inaudible] such as close areas [inaudible]. Maybe they can accept EAI. And always, our system to APP to mobile, always is fairly important for us. There are two point is to promoting EAI with this. Okay.

And after this presentation, I will have a demo we are looking for. Okay, that's all. Thank you.

DON HALLANDER: Thanks very much Marvin. Any questions?

So Marvin you now have, oh sorry.

UNKNOWN SPEAKER: What you said, I must use Core Mail, both sides, sender and receiver to support [inaudible].

MARVIN WOO: No, no.

UNKNOWN SPEAKER: Could you explain please?

MARVIN WOO: If the other side is, such as Google, Microsoft, maybe you also can work. If other sides can't, you support EAI, we are ultimately [inaudible] to change Unicode to Punycode, and we have a solution. It's two account. If the other side can't accept EAI, we can change the alias account to send email.

So unless can work.

ANDREW SULLIVAN: Hi, it's Andrew here. So, this is an interesting trick that you've got there, and I like it. But how do you, how do you handle the case, or maybe the answer is you can't, where you have mixed addresses? So some are EAI systems, some are not. Do you test

them all and then downgrade? Or do you just downgrade when you run into an ASCII only address and you revert to the compatibility address? Or what happens?

MARVIN WOO: Okay. We can't test others of the world, but we do a judgment before we [inaudible] and send email. [Inaudible] and before do this work, the system will be judgment, the system will accept Unicode ASCII. So.

MARK: This is Mark. So, to, I think, further clarify that question, if you were sending an email to, and I think I know the answer to this, but if you were sending an email with two people on the line, one of them was their service supported EAI. So their service advertises, I support SMTP UTF-8, and the other one doesn't, would you downgrade both of them, or would you only downgrade the one that doesn't support SMTP UTF-8?

MARVIN WOO: Okay. I see. Indeed, we can [inaudible], always. I send the email to, always I use my Chinese email address, and [inaudible] is also someone that can support EAI, someone cannot. And our system can ultimately carry classified, if that is safer can accept EAI. I give him Unicode, if the receiver can't, we give him ASCII.

Because our email address has two account. One is EAI, one is Unicode. And the other is ASCII. They all send. And just like me, my Chinese name is [Chinese], my English name is Marvin Woo. Also the same person, indeed. Our email address is the same. One is IDN name, the other is ASCII name, so the system can classify it.

UNKNOWN SPEAKER: I apologize, to keep coming back. Since Mark said you did have the answer, what did you think was the answer? Was it both or was it one…?

MARK: I think the answer is that where EAI is advertised, it will be sent, yeah. They will not be both downgraded, yeah. They'll be downgraded as needed.

RAED ALFAYEZ: Hello. This is Raed Alfayez from Saudi Arabia. I have a question. Do you allow your user to choose their domain name, in their language and the ASCII version, or you do the ASCII [inaudible]?

MARVIN WOO: Can you say it again?

RAED ALFAYEZ: Yeah. Do you all the user, the end user, who have subscribed to your service, to chose their Chinese language, I believe? Do they also choose the ASCII version, the ASCII one?

MARVIN WOO: No, no, no. Before I use EAI account, they can choose Chinese and English name. It's two name, just like me. Marvin Woo maybe can choose, [Chinese] also can choose.

RICHARD MERDINGER: Thank you. Rich with Go Daddy. I have too many questions to go into right now because this is so intriguing, but from the previous question, it sounds like the user explicitly gives the native, the Chinese language and a compatibility address, and that the compatibility address is not automatically generated. Is that correct?

MARVIN WOO: Yes. The alias account is choice. If the user only use one, one account, just Chinese name. So it can be changed to Punycode. It's a very, very strange…

RICHARD MERDINGER: Right. So for the domain, the Punycode conversion works. But for the local part, how do you handle if there is only a UTF-8 local part to get it into a compatible alias?

UNKNOWN SPEAKER: I answer correct, you will understanding. They will switch it to Punycode. So Punycode at IDN. So these part will be Punycode, if the user, yeah.

UNKNOWN SPEAKER: So my understanding, correct me if I'm wrong. My understanding is that you have a different, just a different address. And they're bound internally. So there is not an automatic processing of the left hand side to be a Punycode, it isn't like you take the local part of the address and convert it to an ASCII compatible thing.

Instead, it's another address that you pick. So that you have… The way I hear people say this is, there is an English name and a Chinese name. Yeah. So they're just different addresses, and then they're bound together in the backend as alternates of one another, but there is no automatic processing.

MARK: Actually, okay, so in my experience… This is Mark. In my experiments, while trying to set up some demos, and oh boy, when I send from my Core Mail account, my Chinese name, well first of all, I don't know if I have an English name mapped to that account. I think it has only got the Chinese name.

But when I send it to my Outlook account, currently my service has EAI turned off, and both parts, the domain part and the local part, are converted to Punycode. And so, you know, that's… Yeah. So it works. I send and receive. My software understands it. It's not necessarily expected behavior, but yeah.

The interesting thing is that in the front named portion, it's showing the original characters, but the address itself is Unicoded in both the local part and the domain part.

No, and so that is the experience between Core Mail and Office 365 2013.

MARVIN WOO: Just to give you, yeah, account. So you use only one account. If you give him an alias account, I give you an alias account, okay, there has two account names, because it has [inaudible] I just give you only one account. If I have given you two, you can work…

MARK: So if I have the two, and I send Chinese name, it will land as English name?

MARVIN WOO: Yes.

MARK: So I don't have to choose each time I send?

MARVIN WOO: Need two. If…

MARK: Well, the two accounts are aliased to the same mailbox. But they are two addresses, they just go to the same mailbox. So the question was, if I use name number one, will it dynamically decide to use name number two? Or do I have to choose name one or name two at the moment of composing the email message? I think that is the question.

UNKNOWN SPEAKER: And my experience is if you have to choose.

MARK: Yes, that was my expectation.

MARVIN WOO: When I, email, I need two choice. IDN or ASCII. [CROSSTALK]

MARK: …I choose which of my two aliases.

MARVIN WOOD: Yes. You can set a default account. And either two choice.

MARK: It's very useful. We were just wondering if maybe there was a magic version.

DON HALLANDER: It's Don here, and I just want to make sure that… I have a question, and I'm assuming if I have a question, other people may have the same question. But my understanding is, an EAI compliant transport agent, when it opens a connection to the next hop, says "I'm EAI, or I'm SMTP TF-8 compliant, are you?" And if they get a response, then they say, "Fantastic, we will talk EAI."

But if they don't get a response, then they know that the next hop is not EAI compliant. Does that sound right?

UNKNOWN SPEAKER: Almost. The thing is, when you make the connection, you do a ESMPT connection, and the server advertises it's abilities in ESMPT. So the server says, "I can do this first." And then the client says, "Oh, I know how to do this." Then you can proceed. So it's slightly reversed, but that's the only detail.

The problem is what happens when you have an EAI mail, and the server doesn't advertise the ESMPT, the SMPT UTF-8 extension. The answer, at that point is, you fail. That's the way the protocol works. So the protocol says at that point, not what we just heard which is, you proceed, you do some kind of downgrade.

We tried the downgrade thing. We ran an experiment. It turned out that it just made a mess. So we didn't do that. So the standard says, "Don't do that." And then you're supposed to fail. And the problem in email, of course, is that that's hop by hop. So you can have an upstream mail server that fails, and at that point, the whole thing gets rejected and it bounces back.

And that problem… So it sounds like this solution, what it does, is it downgrades on the first hop, if need be. But if you've got intermediate hops along the way that fail, then you're going to get a bounce. There is no way around that.

DON HALLANDER: So, are there fewer and fewer intermediate hops actually happening in practice? Or are…?

ANDREW SULLIVAN: It's Andrew again. So, this really depends on the kind of email you're talking about and who you are talking to. So in one sense, yes, there are fewer and fewer hops because basically, most of the email on the Internet is handled by five people, or five companies. And they upgrade their service one thing at a time, so there are multiple hops inside of there, but there are, but when a service is upgraded, you get the service one way or the other.

The problem is the tail. And in the tail, there is all kinds of crap. And in particular, in the tail, is also the email sending people, right? The bulk senders, the people who handle, you know, all the fun mail that you get that is non-spam but still bulk.

Well, depending on how you feel on what spam is, but you know, it's the stuff that you actually for, or they think you ask for. And that stuff, there is a lot of crap in there, there is a lot of hops. There is frequently a whole, a long processing chain because they've got to unpick the mail in order to deliver it to you and deliver it to the right place.

And those things, I think, are going to be the place that is most resistant to EAI upgrade, that may not be the end of the world, right? Because bulk email is maybe not the thing that we're most concerned about. But that's the place that I think there remain a lot of hops.

UNKNOWN SPEAKER: I think you bring up a good point when it comes to the bulk email and the types of email that we expect, and it is not spam but it isn't necessarily something that we are target expecting, when the downgrade occurs or the alias is used, and you unsubscribe an email address, which one are you unsubscribing?

Are you, you know, is it the EAI address? Or is it the alias that is being, the vendor is being told to remove from their send list? Do your systems, when you create inbox rules, do they need to be duplicated? Or are they more robust to the degree that they understand EAM? Have aliases. And when you talked about trying this in your test, Andrew, and it made a mess, I think it's a quote. I'm not sure.

I can see where it gets really messy very fast. But go ahead.

ANDREW SULLIVAN: It's Andrew again. So the, it wasn't that I did the test, it was the working group. So when EAI was being developed, there was the first versions and they were experimental. And there was a downgrade path, and there was a downgrade path not just for this, but also for IMAP and for POP and for mailing lists and so on.

And it was those corner cases that got really bad, right? Because what would happen is, you have a case where somebody would, like some people had an EAI address and they worked within an EAI system. And there were other people who had non-EAI addresses, and then what happens when you do reply to all?

And the answer is that everybody's threading just disappears. Like it just turns into atoms. And there were, I mean, the working group spent an enormous amount of time on that topic. And it was just a rat hole, that I recall. Edmund was there too, maybe he feels differently about it.

EDMUND CHUNG: Well, definitely rat hole. But to me, my summary of it is that there wasn't a good way to really discover the alias name. That was the key issue. If there was any way to, you know, ask… Because once you forward an email or you reply a few times and you add a few people in, the problem is you may have

encountered one that is compatible, and then you have no fall back mechanism.

Now you have to discover that alias, then there is no way to discover that alias. So I've personally, I've always said if we have a way to discover the alias, that would be a new command, for the SMTP or whatever, then we can actually do fallback. But you know, that was a lost debate.

UNKNOWN SPEAKER: When you are receiving email, it is quite obvious to receive what venue you are sending email. And how the recipient will receive in Chinese or in English name, so obviously it will receive in English email. And if he or she receives English mail, then what would be the benefit of getting this Chinese email address?

MARVIN WOO: The benefit, I think, for Chinese users like different Chinese users to Chinese users, there are fairly to use, to remember, because the [inaudible] language is fairly easy to remember. Of course, we can't remember so many email address, but it's another languages so easily to remember.

If the email is between a Chinese and another language, because like English, so the different need Chinese name, but he can use

an English name. so the benefit, I think, is for the another language between another language users.

And if there are not the same language, they can also can use English. So there are some compatibility.

MARK: This is Mark. I think to clarify that point, Chinese user to Chinese user. You have 700 million accounts, I think you told me that 50 million of them were using EAI already. And that was almost a year ago, so it must be more. So that's a pretty big critical mass of people who are benefiting right now, just within the Core Mail system, just within that boundary.

So is that correct numbers? Or was it 50,000? I don't know, what's the order of magnitude, you know? No?

MARVIN WOO: Well, indeed we given all or [inaudible] [sass] users and provide EAI [inaudible], but few of them to use it. And we have a plan from, to latest for [inaudible], our most…

100,000.

MARK: 100,000.

UNKNOWN SPEAKER: Actually I think the list of that figure as of late February, that registry users on that Chinese IDN platform is dot [inaudible], is 100,000.

MARK: So potentially millions, today 100,000. So it's still good benefit for anyone who wants to use it, Chinese person to Chinese person.

UNKNOWN SPEAKER: There is a question from the remote chat, from Daniel. He asks, "In my understanding, no matter which email address you send the mail to, it will be delivered to the same account. Is it right?"

MARVIN WOO: Yes, yes.

DON HALLANDER: It's Don here. I'm just aware of time, and we should be breaking for tea, but we should have also done some demonstration, live demonstrations. What I would like to do is break for tea, and perhaps get Marvin and Raed and Mark to have a cup of tea together, and figure out how we might actually see this stuff in action. If that's all right with everyone.

So why don't we take a 15 minute tea break. People can get their pencils sharpened at the coffee machine. And we'll return when the big hand is on the nine. Thank you very much. Can we stop the recording now as well?

So if we could reconvene please?

I never heard that.

So what we're going to do with respect to the schedule that we had, is we're going to save the email demonstration until after lunch, and for those… How many of you have done software development? How many of you have ever done a live demo?

In front of a Board? No.

UNKNOWN SPEAKER: Surely the question is, how many of you have failed to have your live demo work? [LAUGHTER]

DON HALLANDER: So we're going to do that after lunch, either before we review the UASG 007 draft, or when we need a break for light relief. So we'll just move onto the agenda. And Rich Merdinger is going to talk about the work that Go Daddy and VeriSign, the largest registrar and the largest registry have done as they have reviewed their systems for being UA ready. Rich?

RICHARD MERDINGER: Well, thank you very much Don. And Dennis couldn't be here with us today, so I'll be here for both of us. As a registrar, and with VeriSign as a registry, ensuring that the products that we create actually function in the world is extremely important. And we see, especially at Go, from Go Daddy's perspective, the size of the ecosystem that we have with the domains being registered, interactions with vendors such as VeriSign, creating products that need to function, hosting, email, etc.

We pretty much cover the breadth of the UA issues that exist. So it's a daunting task to think that we're going to analyze this. So when Dennis and I started, we took a very simple approach, one of which would be to, as we say, eat your own dog food. And that is, if we're going to be creating domains that are emerging and being rolled out on a weekly basis, may have longer characters, unique characteristics, or be an internationalized character sets, we need to make sure that our own systems can function properly.

And to do that, and think that we could attack the entire Go Daddy ecosystem at the same time, it was too big of a problem. So we started just simply looking at the vertical, that is VeriSign and Go Daddy interacting in the provisioning of domain names,

and looking at the operational components that come along with it.

So the approach we took was necessarily one of a vendor to a client relationship, under NDA, and we talked about everything that we could possibly think that could go wrong, we did it in a closed forum, excuse me, I'm just going to, so I can see the room better.

We did it in a closed forum so that we could have 100% transparent and collaborative discussions. So I can't share every nuance of what we went through, but the intent was to identify core system issues, and the different pieces of the systems that could be impacted by universal acceptance issues, make sure that we engage with our own internal experts.

We have a registration architect at Go Daddy, VeriSign has people that are more technical than myself, and we've got them in the room with us. And we went through some very, very detailed data definitions and some structures that are shared between the two systems, and walked through them, node by node, to understand what the impact UA may have both from how the system inner operate, how they're stored, what the data we were getting from customers was, where the validation needs to occur, things of that level.

So the level of detail that we got into, may often times took an hour, hour and a half, to cover a topic, which may have only been extrapolated out to a broad issue that we could apply to the, not just the industry, but to software development and universal acceptance in general. So while it might have been a specific node in whether or not the data was going to contain one piece or another, we tried to bring it out and say, when dealing with this, you should do this.

And that way it could apply more broadly than just to our one solution. So, we in going through this, there were several components that we felt were involved. System integration, where the registry and the registrar may be doing interactions via EPP, or file creation, file loading, things of that nature.

Once the information is moving between systems, the integrity of the data was essential. So that when something is received on one side, it is, it's what was intended and it's stored with integrity. The customer data itself needed to be stored with integrity. Customer communication became an issue when we started thinking about customers providing us email addresses that may include EAI, or at least IDN.

Will the emails get out to where they need to go? Looking at the 2013 RAA, we have policy compliance issues because we need to be validating that the registrants are who they say they are and

we're using email to do so, and if we're accepting data from them, we have to make sure it works. And then all of this is really underpinning the concept of customer satisfaction relative to universal acceptance, because if we have systems that don't function properly, it will negatively impact customer satisfaction, but just because our systems function properly doesn't guarantee customer satisfaction.

So from system integration standpoint, EPP is already very UA friendly. It is an encapsulating technology that allows us to do whatever we need to do, both in ASCII and UTF-8. In walking through a very detailed node by node data decomposition, we found surprisingly few elements that needed special handling for UA, because the data itself is either UTF-8, or it's ASCII, but they're just data nodes inside the construct.

The most important issues that we ran into came along in host names and labels, because as we sit in an industry and we talk about how our systems inter operate or if you need to know something from someone, you say, what's the domain name?

And it used to be enough to say, using LDH, that it is Go Daddy dot com or VeriSign dot com, but what it turns out now is, are you going to give me the A label version or the U label version? Second most interesting piece here, and it doesn't deal with IDN in any sense, it's just the system configuration complexities that

have come up. We, as in industry, we're used to having a registrar connect to the registry system.

Now with the introduction of so many top level domains, we have registries that are segregating their systems and having farms of TLD SRSs. So now the question is when dealing with this TLD, which of these systems are we connecting to? And we have a new layer of system configuration that we needed to introduce, so that we understand which of those destinations to use. Is this making sense?

Andrew is nodding so that doesn't guarantee anything because he gets this, so okay. When it comes to data integrity, the lesson we learned out of there is not to make assumptions. We talked in general terms, as I mentioned, about domain name, or a TLD, or a label, but you need to be very explicit on what is going to be transmitted between the two systems.

The best practice that we developed in our discussions was that if you're going to have data pass between the systems, the label should be explicitly an A label or an U label, and if that tag is not there specifying which form it is in, then you have a default that is understood between the system. So the onus is on the creator of the file or the sender of the information, to follow the rules of the default, or to at least be explicit on what is sent.

It's not enough just to look for the XN minus minus and know that it is going to be an IDN name that is in its ASCII encoded format. You don't want to make assumptions about what you're getting because the rules can change, then you can end up with deprecated or older logic built in that is now no longer valid, like a TLD is not longer than three characters. That type of, the kind of stuff we're dealing with.

That said, the implementations that we create need expect the rules to be followed, but enforce them and handle them gracefully. It is analogous to the idea of saying EAI email that when you get to a certain point where you run into an email system that will not support the UTF-8, you don't just eat the message and let it die, you actually follow the protocol and bounce the message back as under a little undeliverable. So it's a known outcome.

And something that we caught at the end, and that's probably why there is not a bullet there, that last one, and that is enforcing storage fidelity. And that is, underlying infrastructure may be expecting to handle ASCII characters, and when trying to store UTF-8 email addresses, etc., that's something that can end up, you could end up encoding, developers might encode the data, or they might do something to it, but we really want to see the storage mechanisms actually storing the data that was

provided and not some sort of incarnation of it, just as a best practice.

So customer data is especially fun, because you never know what somebody is going to give you when you ask for something. A great example is phone numbers. We ask for phone numbers and they're well-defined, but we end up with people putting in all zeroes, or they may put in two hyphens because the code was written, they'll look to make sure that there is hyphens or parenthesis, or plus one, or a plus six two, or whatever it happens to be.

So you need to have robust logic built in that is going to take the data that is given to you from the customers, and is going to be able to make sense of it at the input layer so that the systems downstream can handle it better. There is a certain amount of validation that needs to be done, and homogenization of the data, so that we are able to…

I'm thinking in terms again of the validation that goes in with the RAA and needing to be able to identify a customer as a real person. We have real name verification in some regions now. We have, UK had a data quality initiative that dealt with mapping customer data back to well-known sources in the wild. And just when dealing with customer data, you may need to

homogenize it in some way, so that it can be applied to multiple external sources.

And when you do find data that is provided by customers that is outside the bounds and you don't have well-known processes to deal with them, excuse me, well defined automated processes to deal with them, you may need to have an operations group be able to deal with these in a human format, so you're actually dealing with looking up, let's say that you have an email address provided by somebody's system isn't able to discern what it is, or you send an email to verify the domain name registrant and it gets bounced, you may need to have an operational paradigm in place so that you can continue to do so and get the domain validated, or the registrant validated, instead of just failing on the transaction.

And this obviously is communicating with the customers is really where this hits home, because if the… I'm learning so much about EAI, just getting into this process and the conversations today, the idea that someone can provide an email address that has an alias, yet I'm trying to use these two unique addresses now to identify a single entity that is equivalent across the two, are the policies written to support that?

Are they, are the systems designed to bring these two into a single identity, etc.? I mentioned that earlier when it comes to

client side filtering, etc. But so when dealing with your customers, make sure that you're getting more than just email from them. That you have an alternate method to contact them, having an ASCII alias if you're getting an EAI address is interesting, but I think it's not a solution into itself.

So I don't know that ICANN has kept up with the impact that EA, or UA, excuse me, will have on our policy compliance rules, and I don't know if those are active discussions that are taking place in other venues. I'm looking around the room and maybe somebody can comment on that, because if we have contracts written that require us to validate based on email, does that include all email, all future incarnations of email, etc.?

I'm not seeing any comments on that, but these are the types of issues that have, that Dennis and I, when we were going through the data and the policies and the practices that we have were thinking well, how do we deal with this? How do we deal with that? When we were looking at the new IRTP where there is this, the concept of just the change of a registrant now in the WHOIS data requires contacting and getting approval via email, if the email isn't a reliable mechanism, this could have an extremely large overhead and negative impact on the registrant themselves that's simply trying to change the name associated with their domain.

So the impact would be great. Real name verification in China is another good example of somebody that could be effected. So these are things that we haven't really even talked about in this group as far as needing policies to be written in a way that will be resilient to UA issues, and help us in our progress with it.

And then the topic of customer satisfaction. I already touched on this. We sell these products to our customers, we are more than happy to provide an IDN domain name to someone, and we're always happy to sell, we as an industry are happy to sell them an email account that goes along with that, and a hosting platform that goes along with it, but it may not enable that person to accomplish what they want with the product that we just sold them because of non-UA ready systems out in the wild.

And that goes for not just IDNs, but also new longer extensions, etc. I'm preaching to the choir, as we say, because we're all in this room regarding universal acceptance. But if we don't do a good job of satisfying our customers making our systems more robust. There very well could be other technologies that come around the corner, that do a better job of more, that don't have the technical data that the domain industry has, that will provide identities that are more globally accessible and usable, and it's in our best interest to make sure that we are bringing up our technologies as opposed to allowing that to happen.

And the last comment is just that UA un-readiness equals customer dissatisfaction. Again, that doesn't mean that just being UA ready means our customers will be satisfied, we have to do it in a way that actually embellishes their experience, and makes your outline experience better.

So when we were going through the, especially all of the technical and the data, it felt like we were really pretty early in this process yet, but not really. Because some of the issues are so base, it feels like we're very early in the analysis that we're going through. But it's really it is just that there are a relatively few number of major issues that need to be dealt with, but they're very complex and we're trying to deal with it longitudinally so that antiquated or older systems, not antiquated but older systems still function, and the new functionality can still be exploited and used.

IDNs are a reality. We saw this morning that EAI is a reality. Over 100,000 users in China are using it, and we don't want them to only be able to talk to each other. We want them to be able to talk with all of the Microsoft clients, the Google clients, Yahoo, etc. And it's really important to enforce that registry registrar universal acceptance issues are everyone's issues, because if we don't solve them properly, people won't be able to acquire, and leverage, and use the domain names that we are supporting.

So the next step for… I didn't mean to go to Q&A yet, sorry. The next step of what Dennis and I were going to do is get, we wanted to get feedback on the approach. And I know that I covered a lot very fast here, but that we have relatively small set of problems that are complex and hard to solve. We wanted to put this, get feedback on the structure of the deck which is trying to outline the issues not in a very narrow way about specific items, but in the umbrella of the issues that are addressed and faced by the registry registrar group, and take this deck and actually turn it into a paper that goes into the different sections in much more detail.

This is, we're at the very early stages of doing this analysis and it shows, but this is the approach we're taking and I would love to get feedback on whether you think we're missing major points, or if you know, anything you had to say would be appreciated. So there.

DON HALLANDER: Thanks Rich. Eight minutes and 20 seconds. Any questions or comments?

So I do. Andrew does. I'll let Andrew go first because his will be more intelligent.

ANDREW SULLIVAN: And I was feeling the converse. So the compliance issues around email strike me as a potentially thorny problem here. And I'm wondering whether there are maybe two things that may need to be done. The first might be that the areas of the input into, I don't know, the policy development mechanism in ICANN that I can't pretend to understand, that will say, you know, basically, look if you're going to have this policy, you're going to need some baseline of compatibility things, or you're going to have to embrace like this bigger technology, and then you're going to have to have all of these considerations or something.

So that's one piece of it. But the second thing is that for practical operation, I'm wondering if what you need is a recommendation that is basically, look, you need to get a compatibility address from everybody to begin with, just because you know, EAI is not that reliable in a lot of places right now, and then if you're not going to do that, you need to understand the environment that you're in.

So maybe for certain ccTLDs, it's an acceptable sort of trade off because you, normally they've gotten national scope and so you can expect certain kinds of things. I wonder if specific recommendations like that would be useful, or if that just seems like we are wading into, you know, GNSO and ccNSO areas that we don't really want to touch because they might hurt.

RICHARD MERDINGER: Specifically to respond, I would love to say that raising the question to the policy development side of the house, to answer the question for us would help. The issue we have right now is it's so generically stated that these things need to be done in this way that it actually gives us the ability to innovate and make determinations as a technical organization to implement it in a way that can function as opposed to having more detailed policy placed on it, which then we find ourselves almost constrained in the solution set.

And it's not that they wouldn't come up with good ideas, it's just that, well there is business folks that are not in businesses. There are business folks in the technology folks, and you have business analysts and you have architects, and they work independently and together really well.

And each one should be solving the problems that are most appropriate. So I'm not asking that that be done yet, I'm just suggesting that we need to make sure that we appreciate the policy side of this is something I had not, and Dennis had not considered, but is very real.

Don?

DON HALLANDER: Thank you. Don Hallande here. I'm trying to think that the process for getting stuff moving, so you've provided a couple of, a slide deck, and what I heard was basically the EPP, that's fine, some suggested minor adjustments perhaps, but that's pretty okay. But there is more challenging questions about what I would call the demographic information about the registrant.

How do you want to move forward to get this addressed? Do you want to, you and Dennis continue to work on this? Do you want to expand your little working group? Do you want to reach out to the ccNSO, to the registrars, to the registries and say, this is where we've come to and when do you want to do this?

RICHARD MERDINGER: I would very much like to expand out into a broader set. As I started the discussions that we had with VeriSign were very closed and very proprietary in nature. Once those are done, and they're mostly done, but we're not complete, and we've extrapolated our findings from that to serve as a super structure for subsequent discussions, then I would like to expand that out.

But right now, we're only probably 75% of the way through this data model and this data set, but we definitely just want to complete that first analysis, then expand it out to others that can see what we've suggested, have discussions against those

systems, augment it with a broader, with the broader set of UA, and then potentially going out to the broader group.

When I said UASG, that's what I meant to say the UA.

DON HALLANDER: Yeah, so I'm just trying to understand the role that the UASG has in the actions, because it sounds to me like these are Internet industry issues, as opposed to a broader UA issues. But then you bring the EAI issues in, and…

RICHARD MERDINGER: Not to interrupt, but they are Internet issues, Internet industry issues, but the Internet industry is an analog for every issue. For every industry, excuse me, and that's why we're trying to extrapolate them out away from just the registry, registrar, but up to when dealing with domain labels. And granted, if we're talking to American Express, we're not going to use the term labels.

We're going to mention domain names or email addresses, etc., but we want to be able to give them similar best practices of, if someone is signing up for your account, get compatible email address. That maybe a recommendation, I don't know that it is at this point.

And it's the same type of thing we would need to deal with so we could do our registrant verification, they would need to do so for their customer verification. And so the analogs exist.

DON HALLANDER: So next steps from here. What do you think?

RICHARD MERDINGER: Next steps again, finish, get some feedback from this. And I'm Rich at Go Daddy dot COM, if anybody wants to get in touch with me. We may have UASG email addresses some day, we'll see. But basically, Dennis and I finished our analysis, take this deck, turn it into something more robust, and then work with the UASG to potentially, or at least make a request to get some support to build that out into a more formal UASG level document that can be shared and developed and supported.

DON HALLANDER: Okay.

RICHARD MERDINGER: At which point, that can feed into other more externally focused products.

DON HALLANDER: So I see this as a first case study effectively, of an industry and the issues that the industry has come across. So I get that, but it's also specific issues for this industry. And I'm trying to figure out where and how so that we can try to get things happening sooner rather later.

RICHARD MERDINGER: Right now, they're not issues specific to this industry. There are issues that are specific to this one vendor client relationship within the industry, so it needs to be extrapolated to the industry first. Then it can be of the industry, then it can be broader and be of all industries. And that's why we need to finish and…

DON HALLANDER: And just in terms of timeframe, do you have in your mind the June ICANN meeting as the next opportunity? Or there is a GGD summit…?

RICHARD MERDINGER: GGD summit is what we had in mind.

DON HALLANDER: Thank you very much. That's a good answer.

RICHARD MERDINGER: Yeah. And it's one that we actually had talked about beforehand.

DON HALLANDER: Okay. Any other questions?

RICHARD MERDINGER: Elaine?

ELAINE: Thanks. So I just wondered if you could give some examples of your processes to deal with issues that are not automatically remediated. Do you have a story or an example of that?

RICHARD MERDINGER: Yes. We have relatively sophisticated automated validation systems at Go Daddy where, if someone for a specific TLD enters an email address and we need to verify that individual, whether it be that they actually can receive the email, or if we need to be comparing their physical address with a known database doing some sort of verification there, if we're able to affirmatively see that the data that is provided is valid, it can go through in an automated fashion, if it does not, it gets, a queue is populated, and we have representatives that monitor this queue, and they

will reach out to, in the case of Australia, that's the ARBN number, or the ACN number that needs to be verified.

They will go to online resources to do a match up themselves, try to figure it out, etc. So it's basically we have a group of people that handle all sorts of fourth tier domain escalations, and they handle those. Is that what you were asking?

ELAINE: Yes, but also… So if you have a customer who can't be automatically validated and it's an EAI issue, do you generally replace that with, you know, ASCII email? Or what's the usual resolution in that case?

RICHARD MERDINGER: I'd be hypothesizing, but I use to run that department and what I would have told them is get in touch with the customer, or the registrant, via whatever means and make sure that you're able to get valid information that can be validated. So we would request that the downgrade alias, if you will, from them.

Well, thank you very much.

DON HALLANDER: So the next session is looking at UA issues in different parts of the world. Dusan is going to talk about Eastern Europe. Raed

will talk about what's happening in Saudi Arabia. And Lars is going to talk about Western Europe. So Dusan, there is a clicker somewhere, and I will try to get your slides up shortly.

Are you getting them? It would be 6A.

Look at that.

DUSAN STOJICEVIC: Which one? Oh, the bottom, okay. For a moment, I thought you cannot read my name. So basically, I'm Dusan Stojicevic, hard to pronounce like no joke, which we are from the same country, Serbia. And I'm going to talk about Eastern Europe, because I was for past two, three years all over the Eastern Europe.

So, basically I will briefly talk about every new gTLDs and IDNs. We are talking about the region which is originally Cyrillic region. Mostly we are using Cyrillic in our region, but it's not a real case. You will see further.

So basically, when we are talking about IDN and new gTLDs, there is a couple of them. You can see on the [inaudible] com [inaudible] online org [Rus] site, not much. But most of them, except three. They're coming from Russia.

Two of them are owned by Core association, it's from Switzerland, and Bulgaria, and one is dot org public interest

registry. So you know who is it. Basically, when we are talking about Eastern Europe, as you see, you have only one geographic new gTLD, dot [inaudible], or dot Moscow. So we are talking about Eastern Europe, which is not let's say, we don't have….

What is it? Oh, we have speakers.

So, where was I? Ah yes, dot Moscow. We have only one geographic new gTLD. And well, to be honest, this is not a very wealthy part of the world, so we don't have a large amount of money to obtain a lot of new gTLDs.

When we are talking about IDN ccTLDs, you can see that there is… Well, most of them are Cyrillic one, but we have also newcomers like Greek one, oldest in the world, Greek alphabet finally gets through ICANN procedures, and they got dot [inaudible], which means [inaudible] on Greek language.

And we have, for example, Armenia which is, let's say, neighboring to Eastern Europe, or part of Eastern Europe, whatever you want, but they have their own language. Also Georgian is their, with their own language and script. There is a common theme, except Russian [inaudible], which is the most famous in all part of the world, as IDN successful IDN.

Well, we have similar things going on in our IDNs. So at the beginning, we have a large number of registrant, and after

awhile, we lose registrants and those, the number becomes very steady. So basically, what is the problem? The problem is technology for us because all those IDNs are not equally shown to the user, so they look like half of the product, how can I say? Not finished product, so people abandon very easily.

I don't want to mention that we did a survey in Serbia, and the main reason was IDN email, because of the left side, we cannot provide left side Cyrillic, and usual thoughts were, I'm Dusan Stojiicevic, on Cyrillic. I am not on Latin or any, or any other scripts.

So it's a cultural thing, and technology didn't provide good enough solutions for this culture things. Next one.

IDN in second level domain, well we are not only Cyrillic, Greek, or Armenian part of the world, we have Latin characters in our part of the world. For example, [inaudible] Poland. Poland was a pioneer. They've got the complete Unicode inside the table. So from the very beginning of implementing IDNs on the second level. And one that is new is dot [inaudible] for Romania, they I think opened second level domain, IDN in second level six months ago.

And all of those Latin scripts, they're using special [inaudible] for their IDNs. Next.

In second level domain, I think you know that Cyrillic is in dot com, dot photo, dot org, dot link, so Greek alphabet is in this for a long time, so I don't even need to go further with this. In our region, we took actions as a universal acceptance steering group. We were at TLD con, this is ICANN approved top event that is going all over the world, which is organized by Russian registry.

We talked there with a bunch of registries. Domain forum Bulgaria, we talked with a bunch of Bulgarians who are using IDNs, because they have on second level for a long time, but right now they luckily passed also procedures from ICANN after six, seven years. They got IDN ccTLD, dot [inaudible] on Cyrillic.

And we had a couple of events where we talked either privately or on meetings about universal acceptance. Generally, in Eastern part of the world, we have a lot of problems, but we don't have a large software developers or large companies to solve those problems. So we are waiting, except for Russia when we speak. Russia, they have [inaudible] or mail rule or something like that. So basically we are expecting solutions from other parts of the world.

So basically this is from Eastern Europe. Do you have any questions?

DON HALLANDER: Dusan, do you have any sense that mail dot [au] or [inaudible] is actually doing anything?

DUSAN STOJICEVIC: No. No for now. I don't think so. Well, there are aware of universal acceptance, but they're not doing any kind of things like [inaudible] does around here.

ANDREW SULLIVAN: It's Andrew Sullivan here. There is a practical thing that's going on, right? Because the LGR stuff for the root zone is being worked on in parallel with this, and that is going to constrain the root zone in future changes. I guess we already had Georgian, right? That was already processed.

DUSAN STOJICEVIC: Armenian.

ANDREW SULLIVAN: Armenian, rather, yeah. We talked about doing Georgian as the easy case, because it's one language and one script, and it should have been an easy and it turned out to be hard, which is exactly, like that should be our slogan, right? We thought it would be easy and it turned out to be hard. [LAUGHTER]

DUSAN STOJICEVIC: But Greeks are an example.

ANDREW SULLIVAN: So they're working, yeah, exactly.

DUSAN STOJICEVIC: For the hard part.

ANDREW SULLIVAN: They're working with that. So it seems to me that, I know that there are several people who are involved in that work. So presumably, we're getting influence in that direction without it being coordinated here, and I wonder if there might be an opportunity for a little more collaboration.

I know SSAC has worried that, you know, that we're working, and the LGR stuff is happening, and it doesn't seem to be a lot of cross-pollination, so there might be an opportunity there.

DUSAN STOJICEVIC: You're talking about generation panels. So basically I'm chairing Cyrillic generation panel, and it's not as easy as you think, because in our part of the world, the politics is first and second is the job. I don't like this approach personally, but I must be honest.

DON HALLANDER: So just to…

UNKNOWN SPEAKER: …regarding the universal acceptance. So I said, I came a bit late, I want to just ask a question about universal acceptance. And if there is any relation between universal acceptance and intellectual property. But clearing house for newcomers or new domain names, with vis a vis database of trademarks or any IP, or copyright databases.

DUSAN STOJICEVIC: Well, I don't see much relevance, because if you want to obtain the IDN domain name, you have clearing house, everything. So even new gTLDs are going through this process. On the other hand, we need to build policies about universal acceptance, and that is the space that maybe, I'm not sure maybe, we will need to pay attention about IPR.

DON HALLANDER: Can I just intervene. So the universal acceptance steering group, is explicitly avoiding policy wherever we possibly can. We are about the technology, and we are, our target audience is really outside of the ICANN community. It is the software developers,

and to try to get them to make their systems UA ready, so that whatever domain name you want, you can have a fight with the registrant as to intellectual property. That's not [inaudible], we just want to make sure whatever domain name is chosen, works the same as any dot com.

So if you have a domain name, for example, UASG dot technology, technology being a long name, there are places that it does not work, because people have their software says a top level domain name is only two, three, four, six, or seven characters and it is more.

Or you have a domain name that is in Arabic, and they will say, ah no. We don't think that's a valid domain name. Or if you have a domain name that's in Chinese or what have you. So we're actually going through… One of our exercises that we're going through is going through the top 500 or 1,000 websites in the world and by regions, and seeing if you can register for a service with a…

We have, I think, about a half dozen different types of email addresses. And so far, last time I looked, there was one website that worked for all of them. So the intellectual property issues, not an issue ours is focusing on technology. Christian.

CHRISTIAN: That's not to say that brand registries shouldn't take an active interest in what it is we do. It would certainly help their brand and their efforts in capitalizing in their investments, if they put time and energy and efforts into helping us move forward with our goals.

DON HALLANDER: Any other questions? Then we'll go to Saudi Arabia. But we're going to have to upload Saudi Arabia first, so just bear with me. Thank you.

RAED ALFAYEZ: Hello everyone. My name is Raed Alfayez. I am from Saudi Arabia, from Saudi NIC, which is the entity responsible for registering domain names under [inaudible] under the Saudi, so the ccTLD manager.

First of all, I want to thank this group for giving me the opportunity to share with you our experience and initiatives that we have done related to the universal acceptance.

So, as what I have said, we are administrating the same domain space since 1995, [inaudible] since 2010. We are operated by government agency, which is CITC, communication and information technology commission. We are leading the

[inaudible] and regional community for supporting Arabic language and domain names since 2001.

So we have around 15 years of experience. What we have done previously, actually we have started early in 2004, we built test bed pilot project for making domain names available on the Internet, and starting with [inaudible] countries, [inaudible] countries, and then we expanded the project to cover the Arab League.

So it covered 22 countries for the Arab region, and we make domain names in Arabic working before ICANN has open it in 2010. We built many tools, algorithm solution to secure the domain name space, and give the registrant the possibility to manage and enable variance in a secure manner and a simple manner.

We also done lots of IDNS [inaudible]. We have implemented Arabic email project called [inaudible]. And I will speak in details about the last two items. We have done lots of assessment boards. We started in 2007 when ICANN launched example dot test TLDs. One of them was in Arabic, and we have studied applications and see their behavior, and how they support the Arabic language.

As you all know, Arabic language is from right to left. And we have done lots of tests regarding this area. And we publish the

report. In 2010, we have expanded the test, and we covered more applications, and we have sent directory board to Microsoft and Mozilla, for their browser to make sure we had found some issues, and we want them to be resolved.

Our last review board, which was done at the end of 2014, around one year and more little bit, two, three months or something like that. We had expanded the IDN assessment boards to cover many areas, now only browsers or email clients.

So the goal of that, the latest IDN assessment board is to study an address, end user experience regarding IDN implementation for the Arabic domain names. And it covers, as what I have said, many areas and behavior, and we develop a methodology for the test case modeling, and generation, and we have developed online system to capture the result.

So these are some of the areas that we have covered. We have covered browsers, email clients, office suites. And we have used the famous version that have been used in our country, in Saudi Arabia, which was only Windows 7, instead of going all versions of Windows. Android, iO7, [inaudible] 12, and so on. So we cover the browsers, email client, office suites.

Also the search engines, mobile apps, famous mobile apps and content management systems. Some security issues. System

admin tools, web hosting providers. This have all been covered and was part of the score for the assessment report.

So we built our [inaudible] around 190 test cases. And it is detailed this cases, so anyone can go, any user can go and claim it for registers and capture the results, and feed it to the system. So this is the system that we have developed to capture it. So participants will easy have one [inaudible] to capture all results and upload the screenshots that they have faced, and then saved a success story, or success test case or failed test case.

We have published the reports, and at the end they will provide you the result. And it was, you know, only the browsers have the most support for the IDN A. And it was, haven't at least 100 persons, only 60 persons. Email clients, the rest are less than 10 persons. So this shows that there is actually a problem, and we don't want to think about IDNS support, it's only browser support.

No, there is lots of things, lots of applications, lots of areas that need to be studied and have test cases so that developers focus more on them. So these are some examples for the address bar. So we just [inaudible] long domain name in Arabic, so it's a first, dot second, dot third, till eight, dot [inaudible].

And then we tried. So sometimes, most of the times for the [inaudible], most of times it worked, but for some browser,

under some operating system, it failed. And this is one example. If you type this, it will direct you to the search engine. So it doesn't recognize the [inaudible].

This is the hyperlink, so not only the address bar, we tested hyperlinks and webpage content. So if someone click the link, he will get the blank page. It doesn't work. Even a certificate, we're try [inaudible] our certificate and still it says not verified and even it shows the [SSL?] certificate and the Punycode it's not showing it as Unicode representation.

Again, if you simple things, if you just copy the address bar, copy the website, and paste it in a [inaudible] it will be converted. It will not have the same Unicode representation. And this is some problems, and you can see that there are lots of areas that the programmers need to see, not only the address bar if you copy something from address bar.

If you click a link or have inside the HTML body, if you do lots of action, the programmer need to think all about it.

Until now, more than five years in IDNA 2010 haven't been supported. So we cannot use number with domain names, or domain names having inside them. So 2001, I cannot use it until now. Project doesn't support. DNS support it, yes, but the browser does not support it.

They only support IDNA 2003, with the RC4, IDNA 2003. They haven't upgraded to IDNA 2010. This is the office, I believe, one of the tools for office. It doesn't recognize the address, even if you put http slash slash, it doesn't recognize it like the one I showed here under CNN dot COM.

Once you put in www dot CNN dot COM, there will be a [inaudible] recognition that this is a domain, a website, however, for Arabic domain names, it doesn't show, even if you put http colon slash slash, even it doesn't understand it.

In some sessions in support IDN domain names, however in some areas, like if you go to the advanced search, and put the domain name in Arabic, and then click search, it will not, you can see site colon, this is encoding. I don't know. It should be like the exact Arabic name.

Not converted to something, even, this is not Unicode, something else. HTML for encoding. Even system tools like ping, [inaudible], doesn't support IDN at all. This is a problem. This is SSH, by the way, and Bing also doesn't support it. So, this is a summary of the findings for the IDNA.

So IDNA support doesn't mean browser address bar support. It's something more than that. And the result we have got, the user acceptance for the IDN is less than one person. So we need 99

persons more to reach a standard way like ASCII what is, like the support for the ASCII characters.

And it has been 10 years almost, for the publication of the IDN RCs, and still one person, so how many years we need to go more? So we need some, you know, accelerators or you know something try to change, to enforce everybody or encourage everybody to make it support in a good way.

Other things that we have found from the assessment board, there are new aspects, different technologies that will be, that is needed like variance in domain names, variance in email accounts, variance in search engine, how they categorize the website. Also we need some automation. Someone to just start domain name, we need some automation to handle the variance.

From, you know, the registration of the domain name, the variance should be, we believe should be allocated automatically, because it's something the user will not be aware about some variance and why it's not working.

Regarding the email, the [inaudible] project. Phase one of [inaudible] project was started in 2010 until 2013. This was towards a pilot project for testing Arabic email addresses. It was built before the EAI RFCs was published, and it was based on a

hack. So we converted the user part of the email address to a Punycode, and everything was working fine.

We built plug-ins for Outlook, and we changed around the queue to display it correctly. And there are about 100 users participated in that testing, when testing the pilot project. And it is working fine.

The benefit of it, it is working with existing and old email RCs. In 2016 this year, we launched the second phase of [inaudible], and it was built on the new ARFCs, using the standard AI addresses. And we used [inaudible] around the queue, and archive [inaudible], it's difficult to enlarge actually.

Still in beta version. We have successful tests internally with our system, and with Gmail. This is [inaudible] screen for the email that Gmail received from my account at [inaudible].

So what is the finding from [inaudible]? It's almost three years since the ARFCs were published. Until now, there were almost, let's say, no support or very, very limited support in email servers, SMTP, IMAP, POP, email providers, Gmail, Hotmail, Yahoo, because even I put Gmail here because Gmail inbox, the new system doesn't support IDN.

Email address internationalization, EI. And also email clients with mail application, there are very, very limited support.

Another thing, we need protection mechanism for the user part of the email address. And you can see here, two email addresses. On the right side, it's [inaudible]. It is based, the green color here, have three, all in Arabic.

This is the Unicode representation. The left part is exactly mirror of it, but it is different. It is [inaudible]. And there should be some kind of protection for users, so the same user gets both emails or the other emails should be blocked from others to be using, or we will have problems with the phishing.

We need to have automation tools to configure and manage variance, from domain end user accounts. We need actually to boosting the adoption of the new email, the new EIA RFC by high speed service and hosting providers.

Okay, let me, we want to upgrade our system, internal system at Saudi NIC, to support the new ARI. However, we have a SLA with, or we have a service contract with Red Hat, and they said we cannot support the new versions of [inaudible] and the new version of [inaudible] or [inaudible], because wait until it becomes standard and we can have support for it.

And this is actually a problem. So not only high speeds, even software providers or OS providers like Red Hat, they need to make it, support the tools that adopt EAI immediately so that everyone can use it. What is next, actually we believe there

should be a comprehensive study should be done regularly, not only once, regularly to measure the IDN universal acceptance.

Also it should focus on all major players, not only single players all major players, and cover all possible challenges. And recommendations should be delivered and discussed with relative parties. This practice we have started at the Arabic domain name taskforce, and we try to add this, you know, small things that may not be covered when you go from high level perspective.

And hopefully, it's a bottom up approach. And hopefully we will see something coming in the future. If you are interested, you can attend the meeting. I think it's today afternoon. No, it's tomorrow? Okay, tomorrow.

So this is just a sample, a snapshot. I will not cover all of them because we don't have the time, but this is the registrant. And this is are the challenges for the registrant. We have the area of lack of awareness, we have technical issues, usability issues. As you can see, in each one of them, there are lots of issues, lots of problems, lots of challenges that need to be addressed.

This is from the registry and registrar perspective. So the full IDN support and everywhere, lack of IDN support and registration information, security issues, cross registry problems. Even service provider hosting, there are hosting issues. There are

email service issues. DNS hosting issues. Other issues also related to the service providers.

And there is, sorry. Okay, it's just, you know, sample, so hopefully tomorrow we will give you some details about [inaudible]. Okay, that's it.

DON HALLANDER: Thanks very much. That was a lot, very fast. But the slides, I thought, were very clear. And the slides are on the ICANN meeting page. Not those slides, but the previous version. We'll get that fixed. Any questions for Raed?

UNKNOWN SPEAKER: Thank you Raed for your excellent presentation, as [inaudible] also is today. I would like just to ask you about [inaudible] project. If I want to create an email address with Arabic script, is it open for all Arab users? Or just for, is it for a local project for Saudi people?

RAED ALFAYEZ: Well, until now it's a closed project. We haven't opened it. The first phase, it was open for everyone who want to participate. But until now, we haven't opened it, so hopefully we will open it soon.

UNKNOWN SPEAKER: Why is it not open? Is it for, because the other providers are not supporting the IDN email address or there are other issues?

RAED ALFAYEZ: Because actually you can only test between [inaudible] and Gmail. Hopefully Microsoft will be joining soon, and maybe Core Mail then you can send to Chinese addresses. But for Arabic speaker user, only test with himself, so it's not benefit. The previous version of [inaudible], it was working everywhere.

So he can use it and send it, and there are some automatic translation mechanism from Unicode to Punycode, and it will reach. So you can send to any address around the world. But currently it's very limited.

UNKNOWN SPEAKER: Do you have some communication with Gmail? With Google so that they can say that if they can provide some help or support this IDN?

RAED ALFAYEZ: We haven't out date them, but we believe they supported the new AII RFCs, and we hope if everyone support the new RFCs, we

can communicate to everyone. But the problem that's very limited support now from hosting providers, ISPs.

DON HALLANDER: So if I, one of the things that the universal acceptance steering group is doing is creating a camaraderie, bad word, but a group opportunity for people who are trying to do EAI, or who are doing EAI. And the goal there is for people A) to know that the other people exist and are doing stuff, to share some test accounts so that they can look at inter-operate-ability, see what happens, and see what issues come up.

So we know that there are some issues about normalization that you could take different approaches to. And just find out where, what decisions you're making as you do your development, that may not have been covered in the RFCs. You want to talk about Google specifically.

JORDAN BUCHANNAN: Yes, it's Jordan Buchannan with Google. Since Gmail came up, I'll just mention, yeah. So the current state of Gmail is that it is capable of sending and receiving to EAI addresses, but you can't create your own EAI address within Gmail. So as the Saudi gentlemen demonstrated, you can send the email to it, it works

fine. And we're currently in the process of testing actually having EAI addresses as addresses within Gmail.

I would be happy to work with anyone who wants a test account or something like that to play with that, because we'd really like to make it work.

RAED ALFAYEZ: Can I ask you something? If you allow me, this is Raed from Saudi Arabia. I believe inbox doesn't support EAI.

JORDAN BUCHANNAN: I'm not sure the answer to that. You're testing, probably, is as conclusive as anything I'm going to be able to tell you right now.

ANDREW SULLIVAN: This is Andrew Sullivan. So, two things. One very brief, and one more involved. The first thing is, on the issue of variance in the local part, I encourage you to write a best practices document for that, but there is never going to be any specification about how the local part, how variance are going to work there, and that's because it already is weird.

It's already weird. Like, it's not case insensitive, there is no rules at all about the local part. And that's always been true, and in the standards organization, that's always going to be true. So

it's a good idea to write a good practices thing like here are some variance that you would want, or blocking, or whatever, but to expect that to become a standard because A. Sullivan, all lowercase, and A. Sullivan all uppercase can be different mailboxes today.

And that's weird for people, but it's true. It doesn't work that way and nobody does that, it's just part of the spec. The more interesting thing to me though was the point you were making about copying and pasting, and stuff not working at the command line and so on.

Part of the difficulty that we've had, we've always had when we were developing IDNA 2008, we always had the idea that you might actually have a new sort of resolver stack, where you will hand U labels and stuff would just work. And nobody built that. However, the good news is that there is a new API call get DNS, that allows, that gives all the stubs for this.

So if somebody wanted to fund a development effort around, you know, an IDNA aware interface to get DNS, so that it could deal with this, some of the primitives are there but get DNS doesn't have it all. And this is literally intended to be a replacement for, you know, get at or info. So it's all the low level stuff, and you can get everything. And it's got other nice

features like, for instance, user applications can find out the TTL, so you don't have to just guess.

I mean, there is all of these nice things there, but it needs somebody to fund the development for it. Right? People aren't going to do it for free. So that's an example of something where we can say, hey look, here is targeted development that needs to be done. It would support this, and it would be an opportunity to do some real good work that will be basic library stuff that could go everywhere.

Get DNS API, and I think it's get DNS dot net or something. And I know that VeriSign labs has contributed to it some. And I can find you the URIs if you need them. Just catch me offline and I'll mail them to you, or we have some contributors right here in the room now. DNS right? Dot net, that's what I thought, yeah.

UNKNOWN SPEAKER: So we have a question from a remote participant, Andrea [inaudible]. So, the question is, "As I frequently say, one of the fundamental problems is that internationalization slash Unicode slash IDNs etc. is not taught in British university computer science departments. What is the situation with Saudi universities?"

RAED ALFAYEZ: No, we haven't done that. Actually, it's again chicken. We cannot promote or market Arabic domain names, and it is not working everywhere as expected. So we hope that after maybe one or two years, and we have promises like efforts like this group and Arabic taskforce group, to try to promote or emphasize for supporting the IDN in a very good way, so that we can have more domain name registration.

We can do marketing… If you market now, what we give it to people now, they will be shocked that email is not working. There is something called variant. What is variant? So lots of obstacles and we want to solve them before we do the marketing campaign and outreach for universities, and for companies and for local ISPs.

UNKNOWN SPEAKER: Thank you. And we also have an earlier comment from [inaudible]. On a lighter note, today is Mother's Day. There is a Japanese IDN which translates to Mother's Day dot everyone.

DON HALLANDER: Any other questions or comments? So we go from the Middle East, we've heard Eastern Europe. We went south, now we go back north to Western Europe. Lars.

LARS STEFFEN: Thanks Don. This is Lars Steffen. I'm with Echo, association of the Internet industry. Waiting for the slide deck.

First of all, I have to apologize. I didn't use our new UASG template for the presentation. I will do this next time, I promise.

My focus is on outreach, so you can go on with the slides, yeah. Outreach, I'm really looking forward having our documents ready to use them for outreach, because we also have our lists with email service providers, software providers, that we want to get in touch with to raise the awareness and universal acceptance and for this, I think, our documents were at the final stage now are very helpful.

So we have also a new member of staff in the member service at Echo, that I would like to introduce into this topic, to have her onboard to do and support our outreach efforts. The next thing we will have in our pipeline are doing some events. I already mentioned some of them last time. The next one is next week at the WHD global on the next slide, where we have a few domain name talks.

Christian Dawson, who is not in the room right now, will have a panel discussion on new gTLDs. Thomas Rickett, you maybe know from the CCWG accountability, is having a panel discussion on dot brands and one on gTLDs. And Christian and

me will have one panel discussion on the 17th of March about universal acceptance.

And for this panel discussion, I think, on the next slide, you will see, yeah, this is the topic. Building towards tomorrow's Internet, fixing IDNs EAI and universal acceptance. And on the next slide, you will see that we will have a nice setup. Ram will provide a short video introduction to this topic.

He will not be onsite, but we will have Brett [Hoover] from power DNS, the CTO of [inaudible], Peter [inaudible] who can talk about the IDN studies, and the hopefully also about the launch of the Cyrillic version of dot EU. We will have [inaudible] from Core, and you have Christian and me on the panel.

So if you are at WHD dot global in [inaudible], pass by. It will start at 10:00. The second one is the event that I already mentioned last time. It's on 21st of April, taking place in Cologne. It's the so called certified [inaudible] alliance summit, where we will have also a presentation on universal acceptance.

On the next slide you will see that we will be supported by Jean-Jacques Sahel, the vice president Europe ICANN. [Inaudible] the colleague of Mark. Thank you for arranging that. And with me. And we will give the audience, which will be more than 80% of the German email service providers, and 50% of the German ISPs, introduction and overview about our activities and what

universal acceptance is all about, and why it is very important for them as well.

And yeah, these are the activities we will have to our next ICANN meeting. And so more events and outreach is in the pipeline, for example, the last information is on Friday, we also had a small press release that you can see on international dot echo dot DE about universal acceptance.

This session today here of our working group, and that also our members should be aware of universal acceptance. That's it, in short. Thank you very much.

DON HALLANDER: Thank you very much Lars. Any questions, comments? Thanks very much. Very fast.

So, now… What I would like to do is just go through the quick guides quickly, and I'll leave that to Mark. We'll just get the right slides up.

While we're waiting for Mark, I'll just provide a bit of an introduction. We came up with the idea towards the end of last year, so in the middle of last year, we said one of the things we need is a master document with a lot, focusing on the technology.

And last year, we also said we needed a not technical document. And we came up with the UA fact sheet. And so what we've done here is provide something in between, these are quick guides. And the target audience for this is aimed at CIOs and senior system sort of folk to get some idea of the issues that they're going to face when they decide to pursue the universal acceptance issue.

And I'll just, I'll go ahead and go through if that's all right, because you have copies in front of you. So we started out with what the universal acceptance mean. And you saw, first thing this morning, that Ram showed that we had a definition of universal acceptance. That was quite a lot of last year's work.

It's only a paragraph, but to make sure that we were covering all of the things that we needed to cover. So we have, what does universal acceptance mean? This is sort of the overview. On the next page, we break it down. So we add these five verbs. So accepting, validating, storing, processing, and displaying.

And the document just continues and addresses each of those five components. And we provide what we think are good practice guides, not necessarily best practice guides, because that's not, that's harder to define. But we've come up with some recommendations of how you go about addressing universal acceptance.

Making sure your systems are UA ready. So this first page, the page here is, issues to think about when you accept information. The next one is how you validate the information. Do you want to take over from here?

MARK: Well, I thought you were doing a wonderful job, but I certainly can take over.

We're on validate. Validation is interesting, because there are some aspects of processing and storing that are related to validation. The challenge of validation is to determine both whether if something is well-formed, if the string is compliant to RFCs, but also whether or not it's in use on the Internet.

And so there is some well-known techniques for determining, for example, if an email address is alive on the Internet, you simply send a test email to that account. And in the case of EAI, that might be challenging because the systems are not always compatible. So in addition to the issues that we're mentioning here, there are external validation challenges.

And so we're recommending that you do the minimum amount of validation possible. So if you look to see if there are an email addresses is well formed, you know, conforms to a RFC, that's acceptable. But doing much more than that is probably overkill

and will just tend to box you into a situation where you're locking out people who otherwise might be supported.

Now regarding the domain name portions, it's a lot easier. You can query against the DNS. You can do other things like that, it's no different from any other web address. As long as you're not making bad assumptions like things have to be in a certain script, domain names are, you know, a certain length, things like that.

So that last example, that last example is, shows the overlap between validation, accepting, and processing. So during the accepting process is where you may be doing a first round of validation to determine if you think something is well formed, and that's why I say that validation goes across. It's not usually a thing that happens in isolation.

Storing refers to either long term or transient storage of domain names and email addresses. And we recommend that you store things in Unicode wherever possible, and we heard some examples today of what ICANN and Go Daddy have investigated and what they're doing.

It's not always practical to do this, because sometimes there is old code in the system. But if you have the opportunity to do it, you should store in UTF-8.

So processing is actually when you try to use domain names or email addresses. Yes, you may be performing an action like go through a database, collect all of the people who are of a certain format. So we suggest that you use Unicode enabled APIs, that you use IDNA 2008 aware APIs. Sometimes there are distinctions like that.

Do your processing in UTF-8 format whenever possible, I think I already said that. When you have a client server application or a service application, it's a good idea to upgrade both ends simultaneously. It's very easy to get out of synch on one side or the other if you do them in isolation.

And that's a challenge if you have a complex system with many partners, something that was touched on earlier. It's good to get everyone in the same room and agree on what the goal is.

I think displaying is probably the most straightforward. I think we all understand that if there are fonts that are supported on a system, if you see the Unicode points, you should display them, you should render them correctly. Try not to display, in the Punycode format, if you don't have to.

Usually there is no user benefit to doing that. Sometimes there is, but usually there isn't, if there isn't, don't do it. One other point that is mentioned here, sorry Don.

One other point is that email systems very commonly will say, I see that you are using a font that is different from your normal font. This email address contains Cyrillic characters, or something like that, which we think is unexpected. You might be exposed to, you know, confusable characters in a phishing attack, for instance.

And I think that was sort of acceptable in the past. As these new scripts become more and more common, we're going to see more and more examples that fall into those buckets, where they were previously very suspicious looking. And we'll need to come up with some sort of guidelines for when to warn users and what to warn them about.

So there is a Unicode report that you could look at just to get a better understanding of these considerations.

And so the final point is that everyone should become UA ready, and there are a number of ways to do this. I think if you're in the software industry, these are all well understood, but for others who are coming to this fresh, sometimes you need to look at the actual source code to see why things aren't working.

You may need to break the software into chunks, test them as units. And that could be manual, or that could be automated. And if you would like any more detail, there is some here, or you could contact us.

DON HALLANDER: Thanks very much Mark. So I'm quite aware, as we put this morning's agenda together, that this was different then our last two meetings. The last two meetings were much more working meetings, and this morning was very much a reporting meeting and I'm…

And so I acknowledge that. This afternoon is definitely a working meeting, where we'll have a close reading of the next 10 to 20 pages of the UASG 007, which is the comprehensive introduction to universal acceptance. So what I would like to do now, just before we break for lunch, is just go around the room and see if anybody has any issues that they want to raise, any suggestions that they have.

If anybody here is not on the UA discuss list, and the traffic there, I think, is pretty modest, let me know and I will add you tonight. So I just, I'm just going to go visually down, and I'll put on my spectacles so I can see. Just to see if there is any issues that have, no, no.

Kurt?

KURT: This is Kurt. I know we have a UA public session later on in the week, but I wonder, you know, looking at the quality of the

presentations, and there is sort of, you know, good news from Core mail and bad news from Saudi Arabia about the universal acceptance rate, I wonder if it would be too much to take like, make a slide of each one of these issues and present it in a public forum in some way, or you know, get it outside the scope of the people that would attend the UASG session that's public in order to raise in front of the Board, you know, there is real work going on.

And the reporting into the UASG and we're hearing good news, we're hearing bad news, but here is the thing, and it's clear that the whole global community takes this very seriously. That's part of, you know, one brick in the argument to fund UASG efforts. You know, maybe ask for dispensation to get more than, their usual two minutes.

DON HALLANDER: Sorry, are you suggesting at the Wednesday morning UA public session? Or are you saying it at a Board session?

KURT: Yeah, at the big public forum. Not the UASG session which will be attended by all of us who know about, you know, that care about the issue, but instead raise it to the Board and reiterate the significance of what went on, and repeat a small bit of

information from each of the contributors so that they understand what's really been some pretty good, you know, work.

DON HALLANDER: Thanks very much. I'm continuing around the room.

And I'll stand up.

So, I'd like to thank everybody very much. From my perspective, I think the past six weeks have been fruitful, or at least getting ready to be fruitful. We put out a request for proposals in the middle of February for four or five different topics. We received responses. We've got summaries and analysis of that done, and those will go to the UASG coordinating committee for validation, or discussion, and I expect by the time we meet in June some place, I'm voting for Memphis personally.

Then we'll have some significant progress made. So with that, thank you very much, and we'll break for lunch. And we will resume, for those of you interested in the close reading, particularly if you're geeky, and I use geeky in a very endearing term, then we'll meet and I would like to probably, people bring themselves to a smaller area.

Anyway, enjoy lunch. Thank you very much.

So welcome back. I'm trying to figure out how to get the recording happening. There it is, thanks very much. So what I'd like to do is stop the recording from just stop it and not pause it, and then start up a new recording so that we can find it more easily next time.

So this afternoon session is we're going to go through version seven of the introduction to universal acceptance. I think it would be handy if people who want to actively participate in the reading were sort of in this corner, and people who want to just find a table space with Wi-Fi and something to eat can sit wherever they want. I'm relaxed with that.

And for those who didn't join us a couple of weeks ago, Mark is going to just read the document to us, and people can raise their hands to make concerns or comments. and then Mark and Luisa will go back and apply those interventions, and then we should be done. If you say it fast, it's very easy.

So I'm just going to try to get us to the right spot.

Do you remember where we stopped Mark? Page 12?

So I believe that we're on page 13 of 39, and the section called display, and this is in some ways a reiteration of the, what's in the quick guides. And Mark, I'm going to try to give you control of the screen.

Excuse me. Could you make Mark a presenter?

MARK: Am I am presenter because I…? Oh, I am now, excellent.

So point of order, we'll be taking comments as we go. As we go. If there are any comments in the box, Don will call them out and we'll break for them. Display. And if you weren't here for the previous session, this portion is the recommended practices portion. So this is a section about, you know, as we saw in the quick guides, display is one of the verbs, one of the activities that you must do to be universal acceptance ready.

Here are some good practices that are in more detail than what you saw in the quick guides. Display. Display all Unicode code points that are supported by the underlying operating system. If an application maintains its own font sets, comprehensive Unicode support should be offered to the collection of fonts available from the operating system.

Next bullet. When developing an app or service, and there may be a question, should we say app or application? When developing an app or a service, or when operating a registry, consider the languages supported and make sure the operating system and applications cover those languages.

Next bullet. Convert non-Unicode data to Unicode before display. For example, the end user should see example dot [inaudible], as opposed to example dot XN dash dash Q9JYB4C. That conversion is an example of UA ready processing.

Next bullet. Display Unicode by default. Use Punycoded text to the user, I think we should say, display Unicoded, Punycoded text to the user only when it provides a benefit. And you will know your users so you can use your judgment as to whether or not the benefit is delivered or not.

Next bullet. Consider that mixed script addresses will become more common. Some Unicode characters may look the same to the human eye, but different to computers. Don't assume that mixed script strings are intended for malicious purposes, such as phishing, and if the user interface calls the strings to the user's attention, be sure that it does so in a way which is not prejudicial to user's of non-Latin scripts.

Learn more about Unicode security considerations at Unicode dot ORG reports TR36.

Next bullet. Use Unicode IDNA compatibility processing in order to match the user expectations. To learn more, go to that same link as mentioned above.

Next bullet. Be aware of unassigned and disallowed characters. Learn more at RFC 5892.

The next section is on Unicode. This is not one of the standard verbs, but it is a big enough topic that it deserves its own section in this document. Unicode. You supported Unicode enabled APIs. Don't spin your own for string format conversions. Determining which script comprises a string. Determining if a string contains a mix of scripts.

Unicode normalization and decomposition. So depending on what your underlying operating system that you're targeting, the tool change that you're using for development, there are usually Unicode enabled APIs, and it's very hard to do it correctly, so use the ones that are provided.

Next bullet. Don't use UTF-7 or UTF-32. Next bullet. Recognize that mixed script strings will become more common. Do we need that? It's redundancy to the above. I suppose for explicitness, we can have this in this section and the section above, but it seems redundant.

Next bullet.

UNKNOWN SPEAKER: To contribute. I think it's good to have it in two sections because the two sections are very different and specific. So I don't think it's really redundant. It's reinforcing.

MARK: Okay. Next bullet. Recognize that mixed script strings will become more common. Don't assume that mixed script strings are intended for malicious purposes such as phishing, and if your user interface calls such strings to the user's attention, be sure that it does so in a way that is not prejudicial to user's of non-Latin scripts.

Next bullet. Use Unicode in cookies so they can be read correctly by applications. Next bullet. Use IDNA 2008 protocol, and IDNA 2008 tables. Don't use IDNA 2003.

Next bullet. Do not automatically assume that external APIs can consume data that has been NFKC converted.

Next bullet. Maintain IDNA and Unicode tables that are consistent with regard to versions. For example, or that is, unless the application actually executes the classification rules in the table's document, RFC 5892, it's IDNA tables must be derived from the version of Unicode that is supported more generally on the system.

As with registration, the tables need not reflect the latest version of Unicode, but they must be consistent. Next bullet. Validate the characters and labels, only to the extent that determining that the U-label does not contain disallowed code points, or code points that are unassigned in its version of Unicode.

Next bullet. Limit validation of labels itself, to a small number of whole label rules such as, no leading combining marks. Bidirectional conditions are met if right to left characters appear. Any contextual rules that are associated with joiner characters and context J characters, more generally, are tested.

Next bullet. Don't use UTF-16 except where it is explicitly required as in certain Windows APIs. When using UTF-16, note that 16 byte characters can only contain the range of characters from zero to hexadecimal FFFF. And additional complexities is used to store values above this range.

For example, hexadecimal 10000 to hexadecimal 10FFFF. And this is done using pairs of code units called surrogates. If the handling of surrogate pairs is not thoroughly tested, it may lead to tricky bugs and potential security holes.

This is exactly why UTF-16 is undesirable. Linkification. Earlier in this document we defined linkification as the process of a piece of software automatically generating a hyperlink when it

recognizes that a string looks like an email address or a domain name.

Linkification. If a string resembling a domain name contains the ideographic full stop character, shown here, which is Unicode 3,002, accept it and transform it to a dot.

General. Use authoritative sources to validate domain names. Do not make assumptions such as, all TLDs are two, three, four, or six characters in length. That's a common [inaudible] but there are others.

Next bullet. Ensure that the product or features, the product or feature, handles numbers correctly. For example, ASCII numerals in Asian ideographic number representations should all be treated as numbers.

Next bullet. Upgrade your app and server slash service together. If the server is Unicode and the client is non-Unicode, or vice versa, data needs to be converted to each code page every time the data travels from server to client or vice versa.

Next bullet. Look for mail addresses in unexpected places such as, artist, author, photographer, copyright, meta data. Font meta data. DNS contact records. Binary version information. Support information. OEM contact information. Registration, feedback, and other forms.

And my aside on that is, very frequently we will hear from a product or a feature team, my product or feature does not handle email addresses when in fact, they do. They're just in expected places.

Next bullet. Look for potential IRI pads in unexpected places such as, single label machine names, regardless of the loaded system code page, or fully qualified machine names, regardless of the local system code page.

Next bullet. Use GB 18030 for the Chinese language support.

Next bullet. Restrict the code points allowed when generating new domain names and email addresses. All characters that use email addresses, must accept internationalized email addresses, allowing the characters greater than Unicode 7-F. That is, no characters greater than Unicode 7-F are disallowed.

However, an app or service need not allow all of these characters when a user creates a new IDN or email address. Use only the allowed list of characters for IDNs. This is defined at Unicode dot ORG, reports, TR-36, IDN chars dot text.

Some likely security and accessibility concerns can be mitigated by preventing certain [IPNs] or email addresses from being created in the first place. But note, Pascal's law of robustness would still require software to accept such strings if presented.

So our point is, be conservative about what you create, but be liberal about what you accept.

Next bullet. It is important to note that universal acceptance cannot always be measured through automated test cases alone. For example, testing how an app or protocol handles network resources may not always be possible, and sometimes it is best to verify the compliance through functional spec review and design review.

Next bullet. Don't automatically assume that because a component does not directly call name resolution APIs or directly use email addresses, it does not mean that it is then not affected by then.

Understand how network names are obtained by the component. It is not always through user interaction. The following are some examples of how a component can get a network name. Group policy, a L-DAP query, configuration files, the Windows registry, or transferred to or from another component feature.

Next bullet. Perform code reviews to avoid buffer overflow attacks. In Unicode, strings may expand in casing. Unfortunately I cannot pronounce German, so is this [inaudible]. So converting [inaudible] to [inaudible] to [inaudible], when doing the character conversion, the text may grow or shrink.

So in the one example we have four characters, then it converts two different variations of five. If you're buffer was not created correctly, you could have an overflow. And this is also true in Punycode conversions.

Next section. Authoritative sources for domain names. DNS root zone. There are a few options for the authoritative list of TLDs. The first option would be the DNS root zone itself. It is DNSSEC signed, so the list is properly authenticated. You can obtain the root zone at, www dot [Inter-NIC] dot NET domain root zone, or www dot DNS dot ICANN dot ORG slash index dot HTML.

Public suffix list. The public suffix list, PSL, managed by volunteers of the Mozilla Foundation, provides an accurate list of domain name suffixes. This list is a set of DNS names, or wildcards, [inaudible] with dots, and it is encoded using UTF-8. If you need to use the PSL as an authoritative source for domain names, oh, comment?

DON HALLANDER: So thanks very much. There are a number of PSLs. So you might want to just say, the Mozilla Foundation PSL has been the most common.

MARK: Got it. More than one.

If you need to use a PSL as an authoritative source for domain names, your software must regularly receive PSL updates. Do not bake static copies of the PSL into your software with no update mechanism. You can use the link below to get your app, to make your app download an updated list periodically from the Mozilla Foundation public suffix list.

This list gets updated once per day from Get Hub. That's at public suffix dot ORG slash list slash public suffix list dot DAT. As an aside, browsers usually update their internal lists about once a month.

Next section. Other challenges. General. In some applications IDNs are encoded in Punycode as per IDNA, if the name is identified as an Internet name, but UTF-8 is used if the name is identified as an intranet name.

Next bullet. Some older email applications were encoded to a local code page, and they did not have a set mechanism for detecting and converting the character set as needed. This was especially true for the email header. The to field, carbon copy field, blind copy filed or subject.

Next bullet. Some applications that do IDNA, example, Internet Explorer 7, break for non-DNS protocols, and this could affect accessing resources using non-DNS protocols.

Next bullet. Yes?

DON HALLANDER: Just in terms of the word break there…?

MARK: There are a few words I've already seen here like bake, roll your own…

DON HALLANDER: I kind of like bake. But break is just a little, potentially ambiguous.

MARK: Yeah, I suspect these are throwing an exception, but I'm not sure exactly how they are breaking.

Next bullet. When allowing a user to generate a domain name or email address, consider avoiding the use of visually confusing characters to prevent homograph attack. Use only this list of characters for IDN. Unicode dot ORG slash reports slash TR36 slash IDN characters dot text.

Next bullet. When a user is aliasing multiple email addresses, it may be tricky to manage these addresses as a single user identity. Email programs can direct traffic to such aliases to the

same mailbox, but the application may, it should say may, may still perceive these emails to pertain to different identities.

Next section. I would like feedback on this section in particular. Earlier in the document, we defined the term IDN style email. This term has not been well accepted, so we are open to alternates to it. IDN style email and why it is not the same as EAI.

EAI is defined as using Unicode only. A labels, Punycode, are not allowed. Nevertheless, developers have sometimes adapted email software and services to handle IDN style email addresses, rather than to make a full conversion to Unicode.

Because IDNs can be Punycode encoded, some existing software allows the IDN portion of an email address, to be represented in ASCII or in Unicode. For example, some software will treat these two IDN style email address equivalently for all purposes, sending, receiving, and searching.

So the first example, user at example dot [inaudible], and XX dash dash YOUQ53B at example dot XN dash dash Q9JYB4C. For some email clients, those are completely equivalent. However, some software will not robustly treat these addresses as equivalent, even though both are valid, which can result in unpredictable user experiences as messages are replied to or forwarded.

I think we should probably make a note that the user experience can be unpredictable also when emails are searched and sorted. UA ready software and services should be able to handle and treat them as the same. Nevertheless, UA ready software should not generate email addresses that use an A label, they should support true EAI only.

So, we've already seen that part of the robustness of the Core Mail solution is that they do precisely this. When they attempt to send EAI to a service that is not advertising SMTP UTF-8, they perform this conversion. And it works very well, in most cases.

So here we are saying, you shouldn't do it, but we have a solution already that does do it. What should be our comment? I still believe this is the good practice, but we're using words as, you know, should not.

So if there is any feedback on this section in particular, the terminology used in particular, and this last line about should not. I'm looking for feedback from the community on those.

RICHARD MERDINGER: Mark, this is Rich. And the conversations that took place this morning regarding IDN, excuse me, not IDN but EAI, and some of the side conversations that I was part of about the appropriateness of applying the ace algorithm to the local part

and whether or not it would be reliable, I question whether or not this is a…

It may be the best we can do right now, but is it a good practice? And I'm not saying that it's not, because I do not know enough about it. I just know enough to make me wonder. And so as far as what we should say here, I don't know. But we've got people in the room from… And I want to, I would love to give people who know much more about EAI to explain why it's a safe-ish thing to do, totally safe thing to do, or best we can do now so we do.

I mean, I…

MARK: Right. So, yeah. Marvin? So regarding the conversion to the Punycode format, to the Ace conversion, when the… You're sending to a service that isn't advertising EAI, what guidance, oh go ahead. What guidance should we give to the community?

So you have a solution that works today by doing this, and we would like to know, what should we say to the community in a forward looking way? You know, today we know that the ecosystem is inconsistent and doesn't support a lot of things. And so your solution works, and works around that. What should we say is the good practice?

MARVIN WOO: Punycode might be make EAI can work, it's very safe, kind of support. But [inaudible] is fairly bad. Indeed, our solution is to give him tool. That sometimes if [inaudible] forgot the tool, defined the areas of account, so maybe [inaudible] a tool, change it to Punycode. [Inaudible] can work.

MARK: So what I'm getting out of that is our recommendation, a recommendation at this time is, that rather than attempting to do the automatic conversion to a Punycoded, it is a best practice to request an alias that would be valid, that could be used, that you have… That way it's explicit, instead of, or wrong word, but good enough.

UNKNOWN SPEAKER: I think so too.

MARVIN WOO: I think can be [inaudible] is better than [inaudible] is fairly bad.

MARK: It's true. Are there any more comments on that? I think I like this answer. I think I can use that.

Next section. Linkification challenges. Even when applications fully support new gTLDs, linkification of IDNs and EAI might not happen as expected by a user. In some cases, invalid links may even be created. Here are some examples of typical linkification in existing applications.

The first example is, example dot COM. And it's likely that if it's typed in, it won't be turned into a hyperlink. However, www dot example dot COM, will usually change to a hyperlink, www dot example dot COM. Similarly, each TTP colon whack whack example dot COM, can be expected usually to be converted to a hyperlink, but HTTP colon without the whacks will not.

A single label, HTTP colon whack whack dot COM, might change to a hyperlink, but it would be wrong. Example dot news would not change. So this is a case of where the lack of change is not because a new TLD, it's because it doesn't have the leading www.

And so you see below, www dot example dot news, typically would be converted to a hyperlink. Similarly, http colon whack whack example dot news should probably turn to a hyperlink. Example dot photography, no change. There is quite a long list. But www dot example dot photography could be expected to turn into a hyperlink.

Http colon whack whack… Yeah. So, let's see. I really don't want to go through all of this, so…

DON HALLANDER: Excuse me. So Mark, we have some designated use cases that we're getting domains registered for. So it might be useful to use some of those, and then look to rationalize this list.

MARK: That's a good idea.

DON HALLANDER: We've got, from memory there is about 12 domain names with various permutations. So just use those.

MARK: That's a good point.

The comment that I think we should add here, it's not here, is that a piece of software should apply the same rules, regardless whether the email address looks like EAI or not, if the domain name looks like IDN or not. So if the decision is based on the www, that should be the same whether it's internationalized or not.

So as you see in the examples below, China dot com would not change. Www dot China dot COM would change, because it's not looking at those are internationalized characters, it's looking at the prefix.

RICHARD MERDINGER: Excuse me, Mark. I have a question from you. I'm from North America and I see www, so it means the world wide web. Once we've introduced IDNs, is there a set of characters that go along with whatever the IDN is that would be used in its place. Dusan?

DUSAN STOJEVIC: No.

RICHARD MERDINGER: I just want to make sure I'm asking the question.

DUSAN STOJEVIC: I will tell a small story, a little story. When we adopt computers in printing area, usually in newspapers you Cyrillic newspapers, you will get errors like… Which is on the same place on the keyboard where is W. So… dot something dot com.

For example, will be a common error in Cyrillic newspapers. So basically there is nothing similar in IDNs.

MARK: Yeah. And I think we should probably call that out, that these protocols are indicated in ASCII only. And that's not such a problem for browsers today, you don't usually have to type in that part of the string, but for linkification, you know, dynamic creation of hyperlinks, as you can see in these examples, without those prefixes, software usually will not create a hyperlink.

So I think we should add a little bit of text to clarify that.

DON HALLANDER: Edmund has made the intervention. He says, it's not really www and HTTP similar. There are possible translations, but the most popular name is still wwww. However, this does not work with Arabic or right to left scripts.

MARK: If you could send some examples of that. I mean, I'll certainly look, but if you could provide some to start my search that would be very helpful.

Next section. Advanced topics. This is a section that is mainly for developers, you know, people who are writing their own libraries and things like that. So any feedback on whether it's at the appropriate level of detail would be appreciated.

Complex scripts. Right to left languages and Unicode conformance. Most script display characters from left to right when text is presented in horizontal lines. There are also several scripts, such as Arabic or Hebrew, where the ordering of horizontal texts in display is from right to left.

Next bullet. The text could also be bidirectional, left to right, right to left, when a right to left script uses digits that are written from left to right, or when it uses embedded words from English or other scripts.

Next bullet. Challenges and ambiguities can occur when the horizontal direction of the text is not uniform. To solve this issue, there is an algorithm to determine the directionality for bidirectional Unicode text.

Next bullet. There are a set of rules that should be applied by the application to produce the correct order at the time of display, which are described by the Unicode bidirectional algorithm. We generally refer to this as the bi [inaudible] algorithm.

And side comments. Just because this algorithm is defined, doesn't mean that this will be a good user experience. It's pretty hard for people sometimes to look at these mixed direction strings, and know exactly what's going on.

Next section. The bi [dye?] algorithm. The bi [dye] algorithm describes how software should process text that contains both right to left and left to right sequences of characters. The base direction assigned to the phrase will determine the order in which the text is displayed.

It establishes a directional context that the bi [dye] algorithm refers to at various points to choose how to handle the text. To know if a sequence is left to right, or right to left character… Sequences, let me remove character.

To know if a sequence is left to right or right to left, each character in Unicode has an associated directional property. Most letters are strongly typed. I think we have to change the way we use characters and letters in this line to be more consistent.

Most are strongly typed or strong characters, as left to right, as left to right. Letters from left to right scripts are strongly typed as RTL, right to left. A sequence of strongly typed RTL characters will be displayed from right to left. This is independent of the surrounding base direction.

For example, the left to right string, Dubai.

Next bullet. Yeah?

UNKNOWN SPEAKER: Mark, I'm sorry if I missed it. I apologize. In the first line of that bullet, to know if a sequence is left to right or right to left character…

MARK: The word character there should be removed.

UNKNOWN SPEAKER: Okay, thank you.

MARK: Yeah, sorry. And in that paragraph, use of the word character and letter should be double checked to make sure we're using those as defined in RFCs, you know, a comment that Andrew Sullivan frequently makes. It's very easy to use those in a way that are ambiguous.

Next bullet. Text with different directionality can be mixed in a line. In that case, the bi [dye] algorithm produces a separate directional run out of each sequence of contiguous characters with the same directionality.

Next bullet. Spaces and punctuation are not strongly typed as either left to right or right to left in Unicode, because they may be used in either type of script, they are therefore classified as neutral or weak characters.

Next bullet. Weak characters are those with a vague directionality. Examples of this type of character include European digits, Eastern Arabic Indic digits, Arithmetic symbols and courtesy symbols. Punctuation symbols that are common to many scripts such as the colon, comma, full stop, and the no break space also fall within this category.

The directionality of neutral characters is indeterminable without context, indeterminate I think we should say. Some examples include tabs, paragraph separators, and most other white space characters. When a neutral character is between two strongly typed characters, that have the same directional type, it will also assume that directionality.

For example, a neutral character between two RTL characters will be treated as a RTL character itself, and will have the effect of extending the directional run. So you see in this example, the entire string is right to left, and the dot is treated as right to left because of the context, that it's sandwiched between the two.

Even if there are several neutral characters between two strongly typed characters, they will all be treated in the same way. When a space or punctuation falls between two strongly typed characters that have different directionality, the neutral character, or characters, will be treated as if they have the same

directionality as the prevailing base direction, which was defined above.

For example, example… Is that Dubai? Example dot Dubai. Unless a directional override is present, numbers are always encoding and entered big [inaudible]. And numerals are rendered left to right. The weak directionality only applies to the placement of the number in its entirety.

This is a pretty complication section. This is really to the point that, don't write your own. But for people creating their own libraries, this is what you need to know.

Explicit formatting characters are also referred to as directional formatting characters. These are special Unicode sequences that direct the Unicode algorithm to modify its default behavior. These characters can be subdivided into marks, embeddings, isolates, and overrides. Their effects continue until the occurrence of either a paragraph separator or a POP character.

Marks. These characters are very lightweight codes. They act in the same way as right to left or left to right characters, with the exception that they do not have any other semantic effect. If a weak character is followed by another weak character, the algorithm will check for the first neighboring strong character.

Sometimes this can lead to unintentional display errors. These errors are corrected with marks. The mark Unicode 2000E, left to right mark, which is HTML, ampersand, pound, 8206, and this is unreadable. Or Unicode…

I think that whole sentence needs to be rewritten.

Yeah.

Okay. Not hearing any feedback on that, but I think that is uninformative the way it is written.

Embeddings. An embedding indicates that a portion of the text is be treated as directionally distinct. The text within the scope of the embedding formatting characters, is not independent of the neighboring text. Also, characters within an embedding can affect the order of characters outside.

As of Unicode 6.3 embedded is being discouraged in favor of isolates.

So you may encounter this in old code, but don't create it yourself. Instead, isolates. The isolate directional formatting characters indicate that a portion of the text is to be treated as directionally isolated from its surroundings. As of Unicode 6.3, these are formatting characters that are recommended in new documents.

Isolates can also be nested and can be located within embeddings and overrides.

Overrides. The override directional formatting characters allow for special cases, such as for part numbers, to forest a part number made of mixed English and Hebrew characters to be written right to left. Overrides are recommended to override, I can't read that.

Yeah, to avoid… Should be confusion. Overrides are recommended to avoid confusion wherever possible. Overrides can be nested one inside another, and can be embedded in isolates. And so the terms RLO and LRO are used to define overrides. One forces characters to be treated as strong right to left characters, the other forces characters to be treated as strong left to right characters.

POPs. The POP directional formatting characters, but an end to the scope of most recent embeddings, overrides, or isolates. That's shown as a PDF, POP directional format and it restores the bidirectional state to whatever it was before the last LLRE, LRE, LRO, LRO or embeddings and overrides.

Or isolates, sorry.

The bi [dye] rule for domain names. And I think this is important for a registry. So anyone that is allowing someone to define

their own domain name, should be complying with these rules. Section two of the RFC, and I think maybe I should call that un-explicitly who that is valuable for.

Section two of the RFC 5893 lists the following six conditions to be met for labels in bi [dye] domain names. A bi [dye] domain name is one that contains at least one RTL label. One, the first character must be a character with by [dye] property L, R, or AL. If it has the R or AL property, it is a RTL label.

If it has the L property, it is a LTR label. Two, in a LTR label, only characters with the bi [dye] properties R, AL, AN, EN, ES, CS, ET, ON, BN, or NSM are allowed. So question to the reader. These are all terms that are defined within the RFC. Is it valuable to have them listed like this here?

Three…

UNKNOWN SPEAKER: Mark, I was just going to say that without, with that note that you just told us, seems like it ought to be in there. A reference to the RFC for definition or something like that.

MARK: Yeah, it does say that it is defined in section two, but if you're reading this you aren't looking at section two. So, I have some concern.

DON HALLANDER: Perhaps a footnote just to… When people will see a footnote say, oh what does that say? And if the note might say, read the manual.

MARK: Yeah, okay. In a RTL label, the end of the label must be a character bi [dye] property R, AL, EN, or AN, followed by a zero or more characters with bi [dye] property, NSM.

Four. In a RTL label, if an EN is present, no AN may be present and vice versa. Five. In a RTL label, only characters with the bi [dye] properties L, EN, ES, CS, ET, ON, VN, or NSM are allowed.

DON HALLANDER: Mark, this is Don here. So how is this…? I'm with the RFC. Is this just a summary of the RFC, or…?

MARK: It kind of is, yeah.

DON HALLANDER: So I'm wondering if this is even necessary here?

MARK: Yeah. I think it can be greatly simplified. [CROSSTALK]

DON HALLANDER: …my concern.

MARK: Yeah. Let's really whack this one down. Okay. So let's skip ahead.

This is [inaudible], I think we should probably reduce. I'm running off the screen here. Is there any way to make that a little smaller? Or is that on my side? There we go. Okay.

Joiners. RFC 5894 section 4.3. Linguistic expectations, ligatures and digraphs, also known as joiners. Some languages use alphabetic scripts in which a single faux name are written as two characters called a digraph. In other words, a digraph is a group of two successive characters that represent a single sound or faux name.

Common digraphs in English include CH, as in Church, PH as in phone, SH as in shoe, TH as in then, TH as in think. Some diagraphs are fully formed as ligatures. In writing and

typography, a ligature happens where two or more graphemes or letters are joined as a single glyph.

And example is the character AE, as used in English, although that is archaic at this point, in which the letters A and E are adjoined. Another example of a ligature is the ampersand, which was originally an E and a T.

Bullet. If ligatures and diagraphs have the same interpretation in all languages that use a given script, Unicode normalization generally resolves the differences and makes them match. When they have different interpretations, matches must use alternate methods likely chosen at the registry level, or users must be educated to understand that matching will not occur.

An example of this different interpretation can be found in the Nordic languages. In the Norwegian language, the ligature AE is the 27th letter of its 29 letter extended Latin alphabet, which also happens to be the equivalent of the 28th letter of the Swedish alphabet. The same Unicode 00EF character is also part of the German alphabet where, unlike in the Nordic languages, the two character sequence AE is usually treated as a fully acceptable equivalent [inaudible] where the [inaudible] A character.

The opposite though is not true, and these two characters cannot be divided, cannot be combined in a [inaudible] A character.

DON HALLANDER: Mark, so my concern is the level of detail here in specific languages, and I don't think you're going to cover every…

MARK: That is correct. So this is just an example….

UNKNOWN SPEAKER: Excuse me.

MARK: Yes.

CHRIS DILLION: Thank you. I was going to bring up this comment earlier. My name is Chris Dillion, and I'm a member of the Chinese generation panel, and I'm co-chair of the Latin generation panel. There was a line earlier in the document about the GB Chinese code, and then again here, we've got very detailed things about certain languages.

But I wonder if it might be possible somehow to reference the work that's currently being done in the top and second levels. And it rather gets you off the hook.

MARK: I think you are correct. Okay.

So this is another section that I think we can greatly simplify and point to existing resources. So bundling, probably the same category.

DON HALLANDER: So Chris, can you just make sure that you email Mark or the list with the reference?

CHRIS DILLION: I'll do just that.

DON HALLANDER: Thank you very much.

MARK: I think we probably should keep this next section though. Normalization and case folding. But I would like some feedback on the verbiage. Is it efficient? Is it effective? Does it get the point across? Normalization.

Unicode normalization helps to determine whether any two Unicode strings are equivalent to each other. Some characters can be represented in Unicode by several code sequences. This is called Unicode equivalence. Next bullet.

Unicode provides two types of equivalences. Conical, NF, and compatibility, NFK. Sequences representing the same character are called conically equivalent. These sequences have the same appearance and meaning when printed or displayed. For example, Unicode 6E, which is the Latin lower case N, followed by Unicode 303, which combines the tilde.

Or, Unicode F1, which is the lower case letter, I don't know how to say it, [ny-a]. So those are considered to be equivalent even though they are different code points.

Compatibility equivalents are sequences which can have different appearances, but in some context, the same meaning. There is a weaker form of equivalence between characters or sequences of characters. For example, Unicode FB00, the typographer ligature with the two F's, is equivalent to Unicode 0066 followed by Unicode 0066, literally two Latin F characters.

In the example above, the code point FB00 is defined to be compatible, but not conically equivalent to the sequence 66 plus 66. Sequences that are conically equivalent are also compatible, but the opposite is not necessarily true.

There are four Unicode normalization forms. NFD normalization form conical decomposition, characters are decomposed by conical equivalents, and various combing characters are

arranged in defined order. NFC, you know, I think this part here we can probably take out, because we can point to the rules.

So we introduced the concept with these examples, and then once we get into this section with the normalization forms, I think we can probably take most of this out.

DON HALLANDER: Yeah, I concur. Because I think for, I think it could actually scare people off.

MARK: I mean this is in the advanced topic section, but even this is too advanced for this particular document. So we'll give the examples so that people understand the concept and then point them to the defining documents.

So composites, case folding, probably another example. So let's introduce case folding. Case folding is the process of making two texts identical, which different case but are otherwise, quote, the same.

Next bullet. Mapping lowercase A to Z to uppercase A to Z, works for most simple ASCII only text documents. However, it begins to break down with languages that use additional characters.

Next bullet. Unicode defines the default case folding, case fold mapping, for each Unicode code point. There are common and full case fold mappings. The common fold mappings are those which have a simple straightforward mapping to a single matching, mainly lowercase code point.

The full fold mappings are those which would normally require more than one Unicode character. Next bullet. An example of a full case, full mapping is the character… I don't know how to say it. But it's Unicode DF, which is the Latin small character sharp S, a character that is commonly used in the German language.

The full mapping of this character is two ASCII characters S. Some languages need case folding to be tailored to meet specific linguistic needs. One common example of this are the Turkic languages written in the Latin script.

In the classy example, the Turkish word [inaudible] contains both the dotted and dot less characters I. When rendered into uppercase, the word looks like this. Note that the ASCII letter I maps to Unicode 130 Latin character I with dot above, while the letter, the other one, Unicode 131, maps to the ASCII uppercase character L.

Next bullet. One important consideration, according to W3C, I don't think we defined that, is whether the characters are

restricted to the ASCII set, the ASCII subset of Unicode or if the vocabulary permits the use of characters such as accents on Latin letters, or broad range of Unicode including non-Latin scripts that potentially have more complex case folding requirements.

So, probably we can greatly reduce this and go right to the best practices where we point to these resources. So probably, let's look at sorry I'm having a little trouble scrolling here.

I think the first four bullets are probably keepers. And then we should…

I think we should probably mention the W3C in the last bullet, but not this much detail. And then mention that the best cases, the best practices are in this document.

DON HALLANDER: So Mark, what are we doing about identifying words that are covered in the glossary without necessarily defining them earlier in the…? I mean, have we already mentioned W3C?

MARK: I don't think we have.

DON HALLANDER: I know it's in the glossary. So do you want to just…? Louisa is online and taking notes, and I would just suggest Louisa that it gets spelled out the first time we find it.

MARK: Yes, let's do that.

Open issues. This section is a placeholder. So we're very open to feedback on both whether these open issues are valuable, and also whether we have any that are missing. Topics for potential proposals to the ecosystem, ICANN, or the IETF.

Due to differences in IDNA 2003 and IDNA 2008, similar strings such as fooze ball dot DE and foosball dot DE may or may not resolve to the same address. They might not even belong to the same owner. Can or should we encourage bundling at the registration level for compatibility?

That is, should we require a registry for selling both to the same customer?

DON HALLANDER: So this is another, another way of talking about variance. So I think we should use the same words that are used throughout the community and call them variance.

MARK: Variance, yes.

DON HALLANDER: And that's really a policy issue at the registry level.

MARK: So policy issues should be removed.

DON HALLANDER: I think so.

MARK: I think we've established that our scope is outside of policy, so this should be removed.

The next bullet, should ICANN restrict the delegation? Is also a policy question we will remove. Next bullet, defining IDN style email. We've attempted above to do that. The feedback so far has been mixed.

UTR 36 does not discuss structured text. I think we have to go back and put that up. I don't even remember what that one…

Structural separators, defining behavior for bidirectional. I think that in the previous section, where we point to the detailed documents will be sufficient. The last bullet regarding tool tips.

Do tool tips correctly show the TLD for mixed right to left, and left to right, show a visual example and best practices?

So a tool tip is one term for when a piece of software pops up a little UI, either directly above it when you hover your mouse, or maybe it's a banner across the user experience somewhere, that gives the user some additional information about something.

For example, if you type in an email address to Gmail, that contains Latin and Cyrillic characters intermixed, you'll get a tool tip that says, this is a mixture of scripts you may be, you know, this may be trying to trick you. Something like that. Something prejudicial.

So, we should generate some best practices for tool tips related to right to left and left to right. So that would be a situation where it's not someone trying to fool you necessarily, but it's just very easy for the user to create the string in a way that it is, will generate an outcome different than they expect. For all the reasons that we discussed earlier in the document.

Next section, part four, glossary and other resources. Glossary. A label, the ASCII compatible encoded ace representation of an internationalized domain name, that is how it is transmitted internally within the DNS protocol.

A labels always commence with the prefix XN dash dash contrast with U label.

Next bullet. Ace prefix. ASCII compatible encoding prefix.

Next bullet. ASCII characters. American standard code for information interchange. These are characters from the basic Latin alphabet together with the European Arabic digits. These are also included in the broader range of the Unicode characters that provide the basis for IDNs.

Next bullet. API. An application programming interface, API, is a set of routines, protocols, and tools for building software and applications. An API may be for a web based system, operating system, or a database system, and it provides facilities to develop applications for that system, using a given programming language.

Next bullet. Brand top level domain. A brand TLD is an innovative type, let's remove innovative. Is a type of top level domain name, TLD, that is made possible through the implementation of ICANN's new gTLD program.

A brand TLD provides the opportunity for branded corporations to use their corporate name, as their website's top level identifier, instead of a more traditional dot com or dot biz domain space.

So with that example there, traditional dot com or dot biz, should we keep that? Or should we say something about generics?

Is this sufficient?

All right, leave it. Next bullet. CCSLD, country code second level domain. I don't think we actually use that term within the document. ccTLD, country code top level domain. These two letter domains correspond to a country, territory, or other geographic location, for example, dot DE for Germany, dot US for USA.

DON HALLANDER: So, I'm sorry, I just lost it. So these two letter top level domains, and if you want a little bit of trivia, you can say that they are derived from the ISO 2166 list.

MARK: Okay. I think that would be good.

Code points. A code point or code position is any of the numerical values that make up a code space. Did we define code space? They are used to distinguish both number… They're used to distinguish both the number from an encoding

as a sequence of bits, and an abstract character from a particular graphical representation, the glyph.

Let me read that again. They are used to distinguish both the number from an encoding as a sequence of bits, and the abstract character from a particular graphical representation, glyph.

I think if we remove the comma it makes sense.

DON HALLANDER: Mark, did you pick that up from somewhere else? Or did you write it yourself?

MARK: I think Louisa picked it up from somewhere else.

DON HALLANDER: Okay, very good.

MARK: Yeah. Community TLD. I don't think we use that anywhere in the document. So the feedback on whether additional completeness like this, is valuable or not. I'm thinking we should probably remove this.

DON HALLANDER: I agree.

MARK: Next bullet. DNS root zone. The root zone is the central directory for the DNS, which is a key component in translating readable host names into numeric IP addresses. Next. EAI, email address internationalization.

Geographical TLD, let's strike that.

DON HALLANDER: I'm just having the sense that EAI is, should be something other than just…

MARK: Just saying what it means, yes. So, EAI, email address internationalization. Yeah, let's put a better definition there.

DON HALLANDER: And Edmund is saying, if we're taking out brand and community, make sure that we take out any of the other specifics of the new gTLD program, including sponsored.

MARK: Right, I agree. Like a community TLD, that's a policy issue.

FQDN, a fully qualified domain name, FQDN also referred to as an absolute domain name, is a domain name that specifies exact location in the tree hierarchy of the domain name system. It specifies all domain levels, including the top level domain and the root zone.

gTLD. Most TLDs with… Yeah?

DON HALLANDER: Just to clarify. Edmund's point is be consistent, either include them all, or include none. And Edmund, I think we've agreed to include none.

MARK: So, yeah, and I think we removed the references to gTLD elsewhere in the document so we can pull that up.

So how about we just say in or out. IANA? I vote for out.

DON HALLANDER: So do we reference IANA anywhere as the keeper of the….?

MARK: I think we do not. I think we reference only ICANN.

UNKNOWN SPEAKER: There is an explanation of root zone, and we can explain IANA there.

MARK: Okay.

DON HALLANDER: So given the discussions happening in every other room in this building in the next week, you might exchange all of those references from ICANN to IANA.

MARK: Got it. ICANN to IANA. That makes sense. But we'll leave the definition of ICANN here. I think that's appropriate. IDN, internationalized domain names. IDNs are domain names that include characters used in the local representation of languages, that are not written with the 26 letters of the basic Latin alphabet A through Z.

DON HALLANDER: So let me just provide Edmund's intervention. He says, "Is this a sort of, is this a set of glossary or some sort of key terms? It feels to me that we might want a longer glossary that is relevant somewhere, but a short list of the key terms that should be here.

The long glossary could be simply a difference to another link or such."

MARK: Create a separate glossary document.

DON HALLANDER: I don't think that is what he's saying. I think he's saying point to some other place that has a glossary of Internet governance jargon. So this is not the repository of just generic Internet governance stuff. But this is just stuff that we have used elsewhere in the document.

MARK: Okay.

DON HALLANDER: And Edmund says, "Yes to what Don said."

MARK: Well, I understand the concept, but what do we actually proposing to do? Okay, so that still sounded to me like there is a separate document somewhere.

DON HALLANDER: So if… So only include words in the, enters into the glossary that had been used in, used about, but you may also have a, for further reading, look at, see the Internet governance jargon dictionary.

MARK: Okay.

DON HOLLANER: If such a thing exists.

DUSAN STOJICEVIC: Can I ask something? Dusan. Basic Latin alphabet, A to Z, is it ASCII? [CROSSTALK]

MARK: Yeah, we have actually defined ASCII up above, it's a better definition. So we need to align those definitions.

DUSAN STOJICEVIC: And second thing, but I will wait. I will wait.

MARK: So IDNA, I think we should define here. But more detail than this.

DON HALLANDER: Yeah, more than what you have there. Thank you. Because… If you can keep it to a sentence, but then refer to a more thorough document that says, what's the difference? So the people can say, IDN 2003, 2008, 2010…

MARK: Right. IDN ccTLD. I think we should keep this one. Country code top level domain that includes characters used in the local representation of languages that are not written in ASCII. For example, Russia, that should actually be Russian Federation dot RU, Egypt, and Saudi. That needs to be improved.

MAX: One moment. Max [inaudible] for the record. Actually could we change it to IDN top level domains, because there are IDN gTLDs, and from the user perspective currently, is it much difference if you see three letters or two letters? In IDN because in Unicode it will be some kind of mesh from…

So it's not distinguishable by human eyes. And there we could reference to ccTLDs, as country level domains, and gTLDs which use IDNs. So…

MARK: I think we could maybe even go further, since we have already defined IDNs, it doesn't really matter whether they are country codes are not, I think.

DON HALLANDER: At the beginning of a document, you give a bit of a history, and part of that history is the introduction of IDN ccTLDs in 2010. So I would…

MARK: You would keep this.

DON HALLANDER: I would keep IDN ccTLDs, sorry. Yes.

DUSAN STOJICEVIC: But it will create impression that IDN gTLDs what is it? So if we avoid the definition of something [inaudible], not very good.

UNKNOWN SPEAKER: Can I ask you, do we have the definition on IDN? Do we have a definition on ccTLD?

MARK: Yeah we do. It's earlier in the document.

UNKNOWN SPEAKER: I think IDN ccTLDs enough clear for the reader, if you have those two definitions.

DON HALLANDER: And for the record, David has given us a link to the ICANN glossary.

MARK: Thank you. IESG, I propose that we remove this bullet. IETF, I propose that we remove this bullet.

DON HALLANDER: So the IETF is the producer of the RFCs. Do we not use the term earlier in the document?

MARK: Well I guess, in the definition of RFC, we do point to the IETF, so I guess that would mitigate…

DON HALLANDER: I would go with it, I would include it.

MARK: So we include it, okay. Keep it, IETF. Language. The method of human communication, either spoken or written, consisting of the use of words in a structured and conventional way.

[Morph-eme?]. I propose, well if we are simplifying the advanced topics, [morph-eme] is the only place that we are really referring to. That we're only referring it to that topic and we're already greatly simplifying that. So I propose that we remove [morph-eme].

Punycode. I think we should use some of the text from up above. It is a way to represent Unicode with the limited character set of ASCII, supported by the domain name system. It should say, is an algorithm to represent Unicode with the limited character set of ASCII, supported by the domain name system.

DON HALLANDER: And you should include the relevant RFC.

MARK: And the relevant RFC. Punycode is intended for the encoding of labels in the internationalize domain names in applications IDNA framework.

Registrar. A registrar is a company where domain names are registered. The registrar keeps records of the contact

information, and submit the technical information to a central directory known as a registry.

DON HALLANDER: Do you want to call out the registrar as the interface with the registrant, or no, it's the point of contact with the domain buying customer, right?

MARK: What does everybody think?

DON HALLANDER: So do we actually talk about this early in the document? Does that hierarchy of register, registrar, registrant, resellers…?

MARK: We mentioned it in a single sentence.

DON HALLANDER: In which case, do it, but registrar is an organization, or not necessarily a company…

MARK: Organization.

DON HALLANDER: And of course, some registries combine the registrar function themselves, and they can be… I'll try to make it as simple as…

DAVID: Just to jump in. This is David from dot NZ. A registrar is a company where domain names are registered, the domain names are actually registered with the registry through a registrar. So yeah the registrar is the south channel for, but not necessarily where a domain name is registered.

MARK: That is correct.

DON HALLANDER: So the registrar is a link between the registrant…

MARK: …and the registry. The registry is the authoritative master database of domain names, registered in each top level domain.

DAVID: Yeah, correct.

MARK: Okay.

DON HALLANDER: Louisa, have you got all of that detailed there, just to make it simplified?

MARK: RFC. A request for comments, RFC, is a formal document from the Internet engineering taskforce, IETF, that is the result of committee drafting and subsequent review by interested parties.

Script. I think we need to fix this. The letters or characters used in writing. I think it's… This is defined somewhere, but it's the collection of letters or characters used in writing. I'm pretty sure Louisa, that there is a formal definition of this that we should use.

DON HALLANDER: And Louisa, while you're taking notes, just make sure that we are consistent with throughout the whole document of the use of letters and characters. And I would vote for characters.

MARK: Yeah. Second level domain. In the domain name system DNS hierarchy, a second level domain, SLD or 2LD, is a domain that is directly below a top level domain, TLD. For example, in example

dot COM, example is the second level domain of the dot COM TLD. Some domain registries produce a second level hierarchy to a TLD, that indicates the type of entity intended to register a SLD under it.

So now we're talking about policy again. I think we can remove that. Sponsored TLD, remove. UA ready software. Universal acceptance ready software. It is software that has the ability to accept, store, process, validate, and display all top level domains equally, and all IDNs, hyperlinks, and email addresses equally.

There was some feedback on where we say all top level domains, whether we should just say all domain names, but I think it's okay to be explicit about top level domains here. And that's really where the problems have been at.

Unicode. I'd like to, just a style point, where it says it is, just simply say a universal character encoding standard. It defines the way individual characters are represented in text files, web pages, and other types of documents. Unicode was designed to support characters from all languages around the world.

It can support roughly one million characters, and can support four bytes per character. Clean up that bullet.

DON HALLANDER: And just have a reference to the…

MARK: Yeah. Reference to the RFC.

DON HALLANDER: No, to the Unicode…

MARK: Oh the Unicode, yeah, Unicode dot ORG. Yeah.

Unsponsored TLD, remove. UTF, Unicode transformation format. It is a method of converting Unicode characters, which are 16 bytes each, into seven or eight byte characters. UTF-7, pardon me, I would like to not reference UTF-7.

UTF-8 converts Unicode to eight byte characters, eight byte bites. ZWJ0 with joiner is an advanced topic, and I think we can probably move that out.

DON HALLANDER: So did we reference at all, particularly in the left to right script space?

MARK: We did. We didn't review it here today because… So it's in this document, we did not review it today.

DON HALLANDER: Then I would keep it.

MARK: You would keep it? Well, I think that if the portion of the document that it's in is cut, then we will cut it here. If it survives, then we will keep it here, is what I was trying to say. Sorry.

Same with zero with non-joiner.

So, style question. The cost linking to the RFCs could be redundant if we refer to them in the glossary above, but I think that's okay. Yes. Okay? Any objections? Yes, okay, we'll do that.

RFC 3492, Punycode. Punycode, a boot string encoding for Unicode for the internationalized domain name applications, IDNA.

DON HALLANDER: So Mark, I just want to make sure that these are consistent. We have one of UASG documents that identifies all of the relevant RFCs, so I want to make sure that we use whatever language we,

so that they are aligned. And if there is things in this, then we would add it to that.

MARK: I agree.

So the ones we have here, the Punycode algorithm, RFC 5890 through 94, the IDNs, and this is just text taking out of the abstracts, I think.

DON HALLANDER: That's what we did for the list of RFCs.

MARK: Okay. So let's just check this for completeness. 3492, 5890 through 5894. Sorry I'm not scrolling. Right, okay. 6530 through 6533 for EAI.

GB 18030, I think I scrolled too far. I'm having a little trouble here, sorry. Go to, yeah, here we go. Okay.

ISO 10 646 Unicode.

We need to fix the pagination as well.

The GB spec. Okay, then we have a list of online resources. I'm sorry, I'm really having trouble scrolling this.

So, probably we could build this out. This is not absolutely comprehensive, but we have a list of Windows APIs, SharePoint APIs, the public suffix list, the TLDs, Android APIs, that should be IOS, sorry, not OS.

The Unicode security considerations, security mechanisms, the Unicode code planes. I think that should be code planes rather than just planes. Where are we?

DON HALLANDER: We want to add the ICANN glossary.

MARK: ICANN glossary. Definition of URIs, dot net framework…

DON HALLANDER: An overview of Unicode generally, so just pointing to…

MARK: Right.

DON HALLANDER: And I suggest that we make these online resources in alphabetical order.

MARK: Okay.

DON HALLANDER: And in the glossary, do you want to reference M3AAWG?

MARK: Yes.

UNKNOWN SPEAKER: And the question I have, do we reference it at all in the text?

DON HALLANDER: We do not. So, it might be useful to do that when we talk about [inaudible] attacks.

MARK: Good idea.

DON HALLANDER: And [inaudible] says it should be MAC OS and iOS.

MARK: MAC OS and iOS, yeah. I apologize to everybody. I'm really having a hard time scrolling this thing just keeps like catching.

DON HALLANDER: I can.

MARK: Could you? I don't know why I'm having trouble.

DON HALLANDER: In the acknowledgments, if you've made an intervention, either today, during the call in January or the call in February, or through online comments. We haven't gone back and captured any of those yet, but if you had, please stand up and take credit for raising your hand.

So thank you very much. From my perspective, Mark, the next steps are you and Louisa, or hopefully Louisa, will make these adjustments that we came up with today, and apply those. Then I'm going to pass it through a copy editor to look for… Sort of copy editing skills. And then we will publish it as version eight to the community, and seek additional comments.

I am concerned that in particularly today's session, which was a lot of very difficult stuff, we didn't necessarily have…

MARK: Very much feedback.

DON HALLANDER: Not to denigrate anybody's ability, but I don't think we had too much geeky input, and I would like to just see if we can find some folks within the IETF, or Unicode consortium or somewhere who would, once we put it through either say, yes they'll have a review of it soon, or after we make these changes and put it through the copy editing.

My goal is to have this then by the end of March.

MARK: So IETF, the Unicode consortium, who else do you think?

DON HALLANDER: I think the M3WAG, they've expressed an interest…. That we have that covered.

MARK: Got it, okay.

DON HALLANDER: We'll ask the folks at Apple and Google explicitly to go through it as they have big impacts in terms of their operating systems.

MARK: But my goal is to have all of this stuff, a version eight available by the end of March. Do you think that's possible? Today is the 6th, 7th.

UNKNOWN SPEAKER: Today is the 6th. Louisa, will you have all of the edits ready? I will be back from vacation on the 15th or 16th. So perhaps that Thursday, we could meet and go over the edits.

DON HALLANDER: And Andre, I see your question, I'm not ignoring you. So if you guys get this done by St. Patrick's Day…

MARK: I don't know when that is.

DON HALLANDER: The 17th of March.

MARK: The 17th of March. Yeah, if we have it done by then, then you have a chance to get it done by the end of the month.

DON HALLANDER: Yeah. So that will give me two weeks for my copy editor to go through, well, I think will be about 28 pages.

MARK: I think that's going to be pretty tight on, you'll be close, but I wouldn't bet that you would be done exactly at the end of the month.

DON HALLANDER: Okay.

MARK: All right, well thank you everybody…

DON HALLANDER: So Andre says, "With the recommendation, USGB 18030 for Chinese language support, do you mean in addition to Unicode UTF, or in place of Unicode?"

MARK: In addition.

DON HALLANDER: In addition. So that just needs to be explicit.

MARK: In addition.

DON HALLANDER: So if we could just go around, Andre… In the meantime, I'll just go around the room as I did this morning, and just eyeball and make sure that people have had a chance to say if they want to say…

So the room is quiet. So I think we're done with the reading, which is further than I expected to get today. So thank you very much for that. And thank you everybody for the contributions. It's a big document. Thanks to Mark very much and to Louisa for getting this work done.

This is going to be core to our efforts. I think the quick guides were very good. I think we've got a few other quick guides that I would like to see done. I'd like to see an EAI quick guide.

UNKNOWN SPEAKER: And another one came up recently.

MARK: Really?

UNKNOWN SPEAKER: It might have been…

MARK: We've been talking about the EAI one….

UNKNOWN SPEAKER: It might have been a quick guide to linkification.

MARK: Oh yeah. We have been threatening to do a quick guide to linkification.

DON HALLANDER: Okay. In which case, thank you all very much. We'll start the recording. We'll take a 10 minute break or so, and then we'll ask Marvin and Mark to see if we can get some email working. And we will paste in the chat room some IDN email addresses. And people can…

What we would like you to do is send a test email to these addresses, letting us know what email application you're using, what platform you're using, if it's a browser, what browser you're using, what operating system you're using, and what results you get, if any.

And we do promise that we will do a reply to each and every email that does come in over the next day or so.

MARK: We can go on longer than, it doesn't have to be during this session, by the way. You know, you can continue using these email addresses over time. And you know, paste them into the cc fields of other email addresses and stuff, get broad exposure. It's not time limited. Earlier is better, of course, but it doesn't have to be this week.

DON HALLANDER: Right. Thank you very much, and we'll come back when the big hand is on the five. Thank you.

MARK: …application, Outlook 2016. And so what we're going to do, is we're going to start with Gmail. And we're going to paste some addresses into a new mail.

So we have my Gmail address, my work address at Microsoft, my Core Mail account and my [Cell U-NIC] account. And we will say, sending from Gmail.

So off it goes. It will land… Yeah. And it will arrive in the inbox, okay, because that was… My own address was on there, so I sent to myself successfully.

It arrived at Saudi. Oh, I see somebody has emailed me. Who is this? Hi Mark. Is that Raed? Probably, because he has access to this account too. And so it arrived successfully over here at Saudi, and it arrived at Core mail.

So now let's reply, oh. And it arrived at Outlook. So, so far so good. Oh, you can't see the Outlook. Oh, you know what? I know why. Because it's… So over here. It's actually important to see the Outlook. So currently sharing, we'll figure that out. For now, just take it from me that everything worked fine.

Okay. So you can see it on the screen there or you can't? Okay. So now let's reply from each of these.

So let's reply all. We'll say, reply from Core Mail. And now we notice that something strange happened. We have these two Chinese addresses here that are not as expected. We don't know where those came from, and one of those has a bad format, and so the Saudi address got converted to a different address, and the China address got apparently converted to another address too.

So that's a little weird. We'll see what happens, but not expected behavior. Then let's reply from Saudi.

Reply all.

And we'll say, reply from Saudi NIC. I spelled Saudi NIC wrong, that's too bad.

And now we'll look to see what happened.

So we've received the Saudi email over at Gmail. Everything looks, that should have been a reply all. Let me do it again.

So sending from Gmail, reply all. Reply from Saudi NIC. Okay. Back to Gmail. We see all reply here. And there is all of the addresses, so looks great. Gmail seems to be working perfectly.

Now on the Saudi NIC side, they noticed that the Microsoft server, Mark [S.V.?] at Microsoft dot COM, has a problem with SMTP UTF-8. So somehow Gmail was able to work around that. Saudi NIC wasn't able to work around it. I actually think that this is perfectly acceptable behavior, but somehow Gmail has worked around it.

On the Core Mail side, Core Mail did get a bounce from those two email addresses that, you know, those unexpected email addresses. That was expected.

And we haven't received the reply from Saudi NIC yet. Okay, now going over to Outlook. Let me see if I can find a way to show you what's going on in Outlook because this is interesting.

So yeah. It does need to be shared and we were just sharing the browser before. So we stop sharing. Oh, okay. So I share my screen.

Bear with me folks.

Can you see it or is it really small? Excellent. Okay. So here is the original email from Gmail. And you can see up on the address line, that everything is as expected, my exchange account, my Core mail account, my Saudi NIC account, and my Gmail account, all represented correctly.

If I try to reply, we'll see what happens there. Now here is my reply from Core mail. So, I've got those two strange email addresses that used to be the Saudi NIC address, so we've got those. But look what happened to my Core Mail address here. It is being expressed as Punycode in both labels, the local portion and the domain portion.

So that's different than the behavior that we had from Gmail. It works, but as you can see, it's a terrible user experience because I don't know who this is from. It just looks like a bunch of gibberish.

Now when I try to send, when I tried to reply back, I'm getting these messages from my server that I couldn't send to the Core

Mail account or the Saudi NIC account because I don't know EAI. That's what exchange is saying to…

I know, yes. So here I am in Gmail, I receive the reply. Back to Saudi NIC. So Saudi NIC, we already saw that. Back to the message list. Who is this? Who do they have a problem with? Oh yeah, same thing. Are these both the same? Yeah, that was the reply.

And we haven't got the Core Mail emails yet. So I think they haven't arrived at Gmail or at Saudi NIC yet. So you can see this mostly works in a couple of cases, but the experiences are all different. What I'm expecting to see, for the reply from Outlook is that either in Core Mail or Saudi NIC, I actually expect it to arrive.

I've seen it happen where it arrived, even though the service said it couldn't ship it. Again, not an acceptable user experience.

But I've had it actually go through. Yeah, well we can't actually show that right now because nothing has landed.

But you know, we do see some of that. And then, of course, Gmail is working perfectly. So that's my portion of the demo right now. So you can see that things sort of interoperate, but they're on the edge cases.

Everybody is treating the edge cases differently.

DON HALLANDER: So Mark, can you either paste the addresses, or send them to the UA discuss list?

MARK: I will do that.

DON HALLANDER: And say, could people give this a go and report what happens.

So Marvin, are we in a position where we can see your Windows based client in action?

MARK: So I'm going to stop sharing. And now Marvin can present.

MARVIN WOO: Okay.

Now, I think I will show that [inaudible] account. Mark is show commerce Chinese account better. He only have EA account, so I will show the [double] account, [inaudible] account, and earlier account.

Now, I will log in. You can see the user name and domain name is Chinese name.

And this email is my working email. Every worker is in this account. So there are two account. Main account is [foreign language] at [foreign language] dot [foreign language]. The alias account is MW at [inaudible] dot com, the send account.

Now, so I log in, when I log in, so I write an email. A new email. Now, in sender, we can see that is Chinese name. And if I try another, [inaudible] MW [inaudible] dot send. So they are double account. And this is another alias. So there are two accounts that can be used.

When I write an email, I will choice. This email account is my default, default account. So I can use it. Every work, I use my email is my default account. So it can be work. In my inbox, more than so many email. 18,000 more than 18,000 email is in my box, it's my indeed to use it, not the [inaudible] account.

So when I write a new email, and now I will send [inaudible] dot com, email account can't support EAI. So let me see what happened.

Hello. EAI.

Now we send.

Okay. I tried to receive it.

Okay. You can see now I received my email with Chinese name and domain name. Now, this account can support EAI. So, I use my alias account to send email. Now you can see, the sender is NW at [inaudible] dot [inaudible]. So that is a double account to use it.

And now send, receive Punycode email. So it's a double account to work. So when I return, [inaudible] English account. Okay. That's my double account. Now, let me show my Windows PC client.

It's the same account I use. So, when I create a new account. No, maybe now this account is the same, send account. So it can work every day. It's a PC client for Windows. Just usually normal client like Outlook. But it can be support EA account. Every day I use it, every day for my work.

You can see the account name is Chinese name, not English name. Every day I was use client for work. And indeed, I have more [inaudible] client, more [inaudible] client, maybe [inaudible] can show my iPhone.

And a send and receive an email, it's just like web client. You can see. I use it every day.

DON HALLANDER: So Marvin, do you have an English interface? Or is it just Chinese at the moment?

MARVIN WOO: My [inaudible], web client…

DON HALLANDER: I found that, but this is…

MARVIN WOO: This is fresh email, Core Mail fresh mail, not only Chinese, because it's a new [inaudible]…

DON HALLANDER: I will go more slowly then.

MARVIN WOO: Maybe I need sometimes to give you English interface. So it's only Chinese interface for our Chinese users. It can work. And usually I use my email and for iPhone, because now every time, may be more half time I use my email with my iPhone, not a PC and not a MAC. Because every day I would go into every where.

So iPhone is fairly important for me, because a lot of email need my return. So iPhone is fairly important for me.

DON HALLANDER: So Marvin, just to, another quick question. You've provided the software to Thailand and to India. Have you created the interfaces in Hindi and in Thai? Or in English?

MARVIN WOO: Just the English interface, because create a new interface need some, a lot of work. So just give English interface.

UNKNOWN SPEAKER: Maureen just wanted to put the iPhone screen onto the projector. I don't think that's possible without special software.

MARVIN WOO: Maybe we can just have a look. Maybe we can show in this place, in the room. At ICANN show Internet. This is Core Mail's APP, and we can use [inaudible] account to receive and send email. And it can support [inaudible] my Chinese domain name and my Chinese user name, just like PC client.

But either can be used in iPhone. So, the function is [inaudible], and this email is my father sent to me. I just receive it in my phone. So, I have the solution of OS, Windows, PC, mobile, every solution I can support, and [inaudible] [sass] all the solution I have, [sass] can be [inaudible], it actually can be sold now.

Okay. That's all. Thank you.

DON HALLANDER: Thank you very much to the two of you. So what we're seeing is that there are a number of EAI implementations underway and in production. And some of them work as expected, some of them don't work as expected, and we're, our goal for the next little while is to learn what we can learn and see if we can come up with any conclusions.

And the goal is to help the suppliers of these EAI systems to get better and better and more consistent approaches. So that's the UA workshop for Sunday the 6th of March 2016.

And I'll check, make sure there is no other questions or comments.

I'll just check online.

So there being none, I would like to thank everybody for your time and attention today, particularly those who lasted the afternoon. I'm somewhat optimistic for the way going forward, and we'll do something similar in Memphis or wherever the next ICANN meeting is.

So thank you very much, if we can stop the recording and then close down the room. Thank you.

[END OF TRANSCRIPTION]