
SAN JUAN – ICANN GDD: UASG Update
Sunday, March 11, 2018 – 15:15 to 16:30 AST
ICANN61 | San Juan, Puerto Rico

UNIDENTIFIED MALE: ICANN meeting GDD UASG Update, open session, Room 209-BC
(ccNSO) from 3:15-4:30 PM, March 11, 2018.

SIGMUND FIDYKE: Hello. Sigmund here.

DON HOLLANDER: Hello, Sig. Are you going to join us?

SIGMUND FIDYKE: What's that?

DON HOLLANDER: You're obviously going to join us?

SIGMUND FIDYKE: I hope so.

DON HOLLANDER: Okay, very good.

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

SIGMUND FIDYKE: This is the UASG meeting, right?

DON HOLLANDER: It is indeed. We have a relatively small attendance but a quite impressive group of attendees. If people could sit in the front, it would be easier for discussion and dialogue. Our goal here is to provide some update on what the UASG is doing and get perspectives of why people are involved and bothered with the session.

So the big hand is on the 3, and we'll start. This is a public forum for the Universal Acceptance Steering Group to bring the community up-to-date with what the UASG has been doing, is doing, is going to be doing. We thought we would take a different approach today.

If people could sit in the front, it's cozier and there's chocolate at the front. For people in the audience who do make contribution or ask questions [inaudible] we have a choice of the red pill or the blue pill. So thank you [inaudible].

My name is Don Hollander. I'm the Secretary General of the Universal Acceptance Steering Group. The Universal Acceptance Steering Group is all about getting all applications to accept all e-mail addresses and all domain names.

The approach we're taking today is a discussion amongst those who are active participants. Why are they here? What are they doing? And their views – hopefully positive views but not necessarily – of the Universal Acceptance Steering Group.

At the front, we have Andrew. Andrew is a former chair of the IAB. He is one of – I don't know, my wife tells me “geek” is no longer an endearing term, but I'll still use it that way – so Andrew is.

UNIDENTIFIED MALE: [inaudible] geek.com [inaudible] is still [inaudible].

DON HOLLANDER: Andrew is one of the people who has a lot of history around some of the issues that the Universal Acceptance Steering Group is trying to regress. He's very helpful in making sure we don't go down the path that other people have gone to. So thank you, Andrew, for coming.

Elaine Pruis wears many hats. She has been working for registries for a long time. She was involved with CoCCA, with Minds + Machines, and most recently with Donuts. She's currently running a scientific company in a rural western state. She's also one of the UA Ambassadors. We'll talk about the UA Ambassador program in a little bit.

Mark is “Mark from Microsoft” because nobody can spell or pronounce his surname. Interesting, his boss is actually M3, so this might be some code in Microsoft. I don’t understand. Anyway, Mark is one of the Vice Chairs of the UASG and has been involved in the UASG EAI program a lot. He’s the coauthor of our UASG007 Introduction to Universal Acceptance.

In the ether, we have Sig, Sigmund from ICANN IT. Sigmund has been responsible for driving ICANN’s internal UA initiatives.

So that’s who we have in terms of a scheduled discussion, but we’re very keen for this to be a very interactive session.

I’d like to start a little bit with just a brief update, no slides. People who want to know what the UASG is doing, there’s heaps of material at UASG.tech. Prior to this ICANN meeting, we sent out an update on recent activities. If you haven’t seen that, let me know and we can send it to you. If you want to become a subscriber to UA-discuss which is our mailing list, you can do so. I think it’s UASG.tech/subscribe is how you sign up for a newsletter. Do not try to use a non-English e-mail address because I can tell you right now it doesn’t work. But we’re working within ICANN IT to make that work.

We’ve been focusing a lot on e-mail address internationalization (EAI). We have documentation, first draft of an introduction to

EAI has been produced. John Levine's the author, and that has been put out for comment.

We are starting the evaluation of e-mail software and services to see how EAI-ready they are. The first phase of that is to define what we're evaluating and what success looks like and also how we determine which software and which services that we'll evaluate, at least in the first instance.

We've completed a study in linkification. Anybody know what linkification means? So for this side of the room, it's when you're in a messenger or Facebook or Twitter or something and you type in a domain name or a website, it creates a link automatically on the fly. That's linkification. Our goal is to ensure that when people do create links either to a website or to an e-mail address that they are consistent for all domain names and all e-mail addresses.

We don't care if they do or don't. That's not as relevant for us, but we don't want .technology not being linked when .com is, for example. So that study should be coming out early next quarter.

We're working to expand our communications resources. We're looking for a technical manager to manage our technical projects. That's been a challenge finding somebody with the right skillset and right interests, but we continue to look.

We're looking at open source programming libraries to get those ready. We're starting work on an open source awards program working with open source awards organizations and trying to make sure that they include being able to support all domain names and all e-mail addresses as part of their criteria.

We've created this UA Ambassador program. The UA Ambassador program is a selection process. People apply for the role. It's about us supporting them and them supporting us. We're looking for people who understand the topic very well and are passionate about the topic.

I'd just like to start the discussion with Elaine who is one of our first intake of UA Ambassadors. Elaine, why did you apply?

ELAINE PRUIS:

Thanks, Don. I applied because the ability for everyone to be online and communicate is really important to me. I felt as a UA Ambassador I could have some positive effect toward that outcome. It's my desire to spread the UA message and [inaudible] involved in the work that the UASG is doing and it seemed like a good way to do it.

DON HOLLANDER:

Thanks very much. So you too could be a UA Ambassador if you have the knowledge and the passion and the commitment and

the time for the role. There is information on the UASG.tech/helpwanted if you want to do that.

Andrew, this is where I say time to put the spectacles on. I guess the key issue, one of the questions I have is how do you go from the theoretical standards that the IETF sets? And give a bit of history as to when you set the standards up, how you actually did try to make things work in a test environment. And how do we get from there to reality? Thanks.

ANDREW SULLIVAN:

This is Andrew Sullivan. Thank you. That's a trick problem. The IETF that is the standards body that works on both SMTP, the e-mail standards, and IMAP and other related ones and also the DNS which of course is how domain names get expressed on the Internet, it's a voluntary standards organization. This means voluntary in more than one way.

One of the ways that the standards are voluntary is people use them and, to the extent people don't want to use them, then we don't have an army that can force you to use them or anything like that.

The consequence of that is you need to have standards that people want to implement because they get something out of it.

If you have standards that everybody needs to adopt before anything happens, you have a really big problem.

Unfortunately, e-mail and the DNS are both places where you need an awful lot of people to adopt something before anything can happen. I don't know how many people have seen this, but there's this so-called hourglass of the Internet. There are certain things in the middle that everything touches, and that's the waist of the hourglass. Things tend to be ossified around there. It's difficult to make changes. E-mail and DNS are both quite old protocols, so they tend to show up in a lot of different places.

The key thing in both of these protocols has been to try to make it so that the people who get the benefit are the ones who have to do the work.

In the case of domain names in the case of IDNA, the idea is that if you want the benefit of having this name, then you have to run your website or your e-mail server or whatever in a way that supports that kind of stuff. If you want to be a consumer of that, then you have to be somebody who adopts the new clients and so on that can work with this.

Then similarly with e-mail, the idea is that you need to have an e-mail server that supports this kind of thing and you need to be talking to people who are interested in it so their mail services are going to be compatible with that.

It's a little bit easier in the case of e-mail, the idea was, because in the case of e-mail most of the time the communications are between people who presumably both speak the same language. It's very hard and not very useful to send an e-mail to somebody who doesn't speak the same language as you. The idea was that end-to-end, you would have a path through that would cause people to cooperate.

The difficulty is that both of these protocols are designed in such a way that in order to get widespread use of them, a lot of other people need to be able to accept that. For instance, if you have an e-mail address that uses Unicode characters, that uses something beyond the ASCII range, then anybody who is accepting that e-mail address also needs to be able to accept it. You want to sign up for some service on the Internet. You need to have that service provider also accept that thing.

I think that's the hard part on the Internet, to get everybody else who isn't interested in this, give them some kind of carrot. I like to say this all the time, so those of you who have heard this before, I'm sorry. On the Internet, there aren't any sticks. There are only carrots. You can bring the donkey along with carrots, but you can't beat it because you don't have any power over anybody else. It's their network. That's the whole goal here, make this enticing to people so that they understand there's some kind of value in it.

When we tried this originally, one of the things that the IETF can do is try experimental protocols first. The IETF had a bunch of experimental protocols for e-mail internationalization. We ran some experiments to see whether it would work, and the first go at this didn't work very well except for the parts that finally became the standards track documents.

There were other things that people wanted that would have been maybe desirable. Like for instance, a way to make internationalized e-mail and uninternationalized e-mail or internationalization oblivious e-mail or whatever you want to call it, to make those work together. But it turns out you just can't. We tried it. It doesn't work.

We tried those kinds of things. You try those things on the Internet and then when they don't work you say, "Okay, well, that doesn't work." Unfortunately, it doesn't work so then in order to get people to upgrade, you actually have to entice them. "Hey, there is this advantage."

We do have an advantage on the Internet today, and I want to point this out because this is something that I think the Universal Acceptance Steering Group has been very good at promoting. That is when a lot of these protocols were designed and built, to give you an idea about the DNS or e-mail, both of those protocols originally when they were specified, you could

get a list on paper of all of the humans who were connected to the Internet. So these are ancient protocols, stone tools time.

Today, systems automation has made updating of your software much less painful than it was back in the day. It used to be this manual process and you go out and so on. Nowadays, we have all kinds of tools, things like Docker and the rest that allow you to try these things out very quickly. Try something out, see if it works and if not, you can roll back.

I think this group has been very good at promoting those kinds of strategies and trying to make sure that people who need to update their software are getting some kind of benefit out of it. One of the things is, “Hey, look, bring your software deployment standards up to the modern thing. It reduces your workload, and while you’re at it you get this benefit that you’re more compatible with the rest of the Internet.” I think that’s a great thing that this group has been doing.

DON HOLLANDER:

Thanks, Andrew. Mark’s saying, “What’s my question going to be? What’s my question going to be?” Mark is from Microsoft. Has anybody here heard of Microsoft? They make Xboxes apparently. I don’t know what that is. Mark has been involved in actual [device] production and development and breeding, I guess.

Mark, Microsoft is a big company and you have very complex systems. The story I heard is that there's 100,000 places in Microsoft's code where domain names or e-mails might be touched. That's a massive undertaking. Why is Microsoft here?

MARK SVANCAREK: I think that's a made up number, 100,000. I don't know where you got that number.

DON HOLLANDER: On the Internet.

MARK SVANCAREK: On the Internet, right. But certainly it is a big number. We have a lot of code, and a lot of it is old. Not stone tools old, but early bronze age old. Software being what it is, everything is possible. People get excited, "We should definitely do this!" So sometimes we have to ask ourselves, is there really going to be a return on this? What's the purpose? Does this drive the use of something else? Does this bring in money? Does this meet a regulatory goal?

In the case of universal acceptance, some of it was very clear. We knew our browsers had to become UA ready because IDNs were already real. Regarding EAI, it was less clear because our

engineers for the most part are English speakers. They come from English-speaking backgrounds, whether it was because they grew up in an English-speaking country or because that's what they were exposed to during their education, and they tend to associate with English-speaking people.

They're attracted by shiny objects just like everybody else. They say, "Well, don't you know e-mail is going away? Everything is going to be WeChat. And passwords, it's all going to be gone next week, and all of this is done."

Really what it came down to is we have a new CEO and we have a new mission statement. The mission statement is "empower every person and every organization on the planet to achieve more."

So you look for places where people are blocked because they don't have broadband access. So we invest in white space technology. Their language is not well-represented on the Internet which means that they have low confidence and a lot of friction when they try to use the Internet. Well, let's look at localization, globalization. How about this EAI thing? Does that fit in there too?

Then once we started thinking about it, then we went around and we canvassed the engineering teams. "What do you know about these standards? How much have you done?"

It turns out that in pockets around the company, a certain amount of the work had already been done either by passionate people who just felt it was important and fought for those features or in some cases like Windows – Windows had already supported EAI and IDN, and Active Directory which is built on Windows Server already support these things because that part of the company had decided that there would be common engineering criteria and there would be a list of things that were supported even if there was no immediate business case. We just felt like that was good hygiene to have.

As we looked around the company and began to canvas, what is the level of effort it will take to support things, we decided that universal acceptance was something we wanted to get behind. They were willing to allocate a small amount of my time to working on it both here at ICANN and also coordinating engineering efforts around the company.

Our phase one efforts are now complete. We've just announced that Office 365 and Outlook.com can support sending to and receiving from EAI addresses.

DON HOLLANDER:

Thanks very much. In the ether, we have Sigmund from ICANN IT who is actually driving universal acceptance through the ICANN IT operation. Anybody here use any ICANN IT services? Maybe

you subscribe to an ICANN mailing list. Anybody use their accounting system or their registry management system or their registrar management system or their ALAC system or their GAC system? That's about five. Sigmund, I think you have another 80 on your list.

SIGMUND FIDYKE: Yes, at least.

DON HOLLANDER: Tell us how and why ICANN is addressing the universal acceptance issue and how you started out and how you changed and why you changed.

SIGMUND FIDYKE: Okay. Well, ICANN's passion is to get everybody on the Internet and not just the people who have an English keyboard or an ASCII keyboard. For us, getting to universal acceptance is a natural fit and something that we want to support in every way we can because that's the rest. Half the world would be more comfortable dealing in their own language like you said, Don, use their own name.

That's our reasoning. That's the mission behind what we do here. As a nonprofit [with] limited resources, we started out just

trying to get every service we have to be fully UA ready, and it took a bit of work.

We took some steps along the way. One was to started with long ASCII because there are a lot of those out there and it was very easy to test. We're moving on through to doing complete UA readiness.

But for us it has been very interesting and getting to the solution has been a very useful task in that we are also getting the rest of our house in order from an enterprise architecture point of view, which is also one of my duties at ICANN IT. Which means we now have a library or a dictionary of every data element we have. We have a better accounting of all of our systems. Getting to UA readiness has been both a driver and a reason to do some real clean up – clean up is the wrong word – but more accounting of what we have and where it all is.

As I get to that and talk to other companies, I found it interesting and maybe a little bit gratifying that we're not the only ones who didn't know – we do now – every service we have and everything we're working on.

We've had an amazing set of other positive side effects, but the main reason we're doing this is because we want to go to where everybody is so that they can get on the Internet and use its features. Thank you.

DON HOLLANDER: Thanks very much, Sig. So you started off with a hiss and a roar, and you found it was really hard and so you said, “We’re just going to take it in phases.” Does that sound about right?

SIGMUND FIDYKE: Yes, it does. Just let me add to that. In doing it in phases, it also helped really define how much work we actually had to do. Where before, the estimate was “a lot.” Now it’s not “a lot” anymore. It’s a number and a roadmap.

DON HOLLANDER: Thanks very much. We had a session with the GAC earlier today, and we got asked that question: “Is UA a black-and-white issue or are there shades of gray?” I don’t know that I named ICANN IT, but we used you guys as the example that said, “There’s a continuum.” You start when you’re definitely not UA ready, which is almost everybody in the world. Then you move to become fully UA ready, which is hardly anybody in the world.

The approach that people like you are taking is, “We’re going to do the low-hanging fruit. We’re going to do new ACSII and long ACSII but accepting all the ACSIIs. As you do that, you get this inventory of all your data fields.

Now I don't know how many people here have been or are system architects, but this is something that system architects all should have and precious few do have. So I think that's a core element.

Those are introductory remarks from the people on the panel. I'm happy to have people from the floor come up to the microphone, or we can move the microphone to you. Remember, we do have the red pill or the blue pill if you make a contribution.

In the meantime, Andrew, tell us about IDNA and is it still stuck with Unicode 6.x when Unicode is already at 11, and how do you unstick it or does it need to be unstuck?

ANDREW SULLIVAN:

Yeah, so a number of years ago – how can I put this? I'm trying not to bore everybody with a lot of inside baseball and at the same time give you enough information to understand what I'm about to say.

IDNA depends on Unicode, and Unicode is the way in which all of the characters in the whole universe are to be encoded. It turns out that there are versions of Unicode, but you don't know what version of Unicode you're using. This is a critical piece of this. The version of Unicode that you have in your machine is the

version of Unicode you have. Each version of Unicode has more characters than the last set, but you can't tell which version of Unicode you're using at any one time.

The original version of IDNA was pegged to one version of Unicode and that, of course, was no good because the next time new computers came out everybody had the wrong version of Unicode and it wasn't working. And there were a number of other problems, so we got the latest version of IDNA which is this mechanism for encoding domain names that have more than ASCII characters in them.

We thought we had written it in such a way that it was completely Unicode version agnostic, that it didn't care. It depended on a number of properties of the characters of the Unicode code points. But it turned out that the engineers who worked on this in the IETF are networking people. They're not language people. They're not character encoding people. The people at Unicode who work on character encoding people, and they're not network people. It turns out that there's a very, very, very, very tiny overlap between people who know something about network stuff and people who know something about character encoding, and so we made a mistake.

It was just a misunderstanding of the different purposes of this because Unicode is really designed for display, and it mostly

depends on the idea that you know what the language is of the people who are talking to you. So you get a document, for instance, or a web page and it says what the encoding is. It tells you what language it's in.

But in the DNS, in identifiers on the network, you don't have that. You don't have any information about it, and there's no way to send it. This is a very old protocol. There's no way to send along, "Oh, and by the way, I'm speaking Estonian" or a better example, "I'm speaking Ukrainian versus I'm speaking Russian." You can't find that out, and so we had this problem.

When Unicode 7 came out, we discovered that there was a code point in there that we thought was impossible and it turned out was a new letter. Well, this led us to do some digging, and we had a little problem. Because it turned out that this was worse than we thought and it had always been there and there's a bunch of problems.

Now what has happened is we've had a little seizure where things are locked up and we don't know how to fix this. The discussions that are going on right now among protocol people are basically that this has been locked up for a long time and we don't know what to do next.

What I think is going to happen, but I can't promise, is that this is going to come unlocked and we're going to figure out another

way around it. We're going to figure out a new workaround. But that new workaround is going to be done at the level of policy rather than the protocol that's underneath it. We don't know whether this is going to work. We haven't worked out all of the details of it yet. But I think it's going to come unlocked.

I think an important thing to remember about this though is that since 2008 or thereabouts when all of this was encoded, there haven't been any characters added to the Unicode repertoire that are the sorts of things that you really are likely to want to use. They're unusual languages. They're new encodings for languages that are used in parts of the world that weren't on the Internet when they encoded things before that.

This is not a place where you're going to have a giant demand for new top-level domains. You're talking about linguistic communities that are very, very small. The practical problems for those languages are not domain names, for instance. The practical problems for those languages are getting enough language speakers together to provide a complete way of writing the language down. That's the kind of situation we're talking about.

Even though this has been frozen, it hasn't been frozen in such a way that it's really affecting anybody in a practical matter, even

though it is affecting people in a way that would prevent certain kinds of names from working the way we would like.

I say it that way with the knowledge that, of course, to the people whose needed code point is not accessible to them right now, that's potentially a problem. But these are identifiers on the Internet. I think this is another really important thing to remember about this. The goal of IDNA is not to allow you to write Tolstoy in the domain name system. It's to allow you to get a good enough identifier so that you can use it in your linguistic environment, in your writing system.

Domain names are not words. They're not paragraphs, certainly. What they are is mnemonics. They're names for things. That's the reason they don't have to be words. You see domain names like w3.org. It's the domain name of the World Wide Web Consortium. Where I come from at least, neither "w3" nor "org" is a word. They're just mnemonics.

I think that this is an important thing to remember about this. The idea is to make something that's good enough to be usable for most people. But good enough to be usable is still a matter of some difficulty, and we have to move forward to make that work.

All of that was a long explanation of the short answer, which is I don't think it's going to remain locked forever, but there's still

some delicate picking through thickets that we have to do in order to make this a little bit smoother.

UNIDENTIFIED FEMALE: Thanks. I want some blue chocolate. Thank you. That was fascinating to me. Can you tell me which letter caused the seizure?

ANDREW SULLIVAN: It turns out there's a series of letters in Arabic script – and remember, Arabic script is used to write a lot of languages just like Latin script is not used exclusively to write Latin. In fact, very few people write Latin in Latin script. We have this Arabic script, and it's used for a lot of different languages. It turned out that historically, there were a bunch of ways that it was encoded that worked best for languages that were dominant historically.

There are a bunch of languages that appear mostly as far as I can tell to be dominantly African, but I'm not perfectly sure about this, and the particular character that triggered this for us was a thing called Beh With Hamza Above. But this is a combination of various kinds of things.

A normal thing to happen in Unicode – now you're going to get way more information than you ever wanted to know, but I'm sorry. Once you open this tap, it's hard to shut. The way that

characters work in Unicode, there are two different ways you can do things. You can either have individual little pieces and you put them together and that's combining, or you have a precomposed character.

A familiar example in Latin script is “e” with an accent on it (é). You can write that one of two ways. You write the “e” and then you write the accent and it's a combining accent so they appear together on the page. But this is a composition character. Or else you have a precomposed character that's also defined in Unicode, and it's the “e” with the accent already on top of it and it's one code point.

Formally, these are the same character as far as anybody is concerned, so there's a step called normalization. You run this through the normalization thing, and out the other end says it's always the same character.

There are other characters when you put them together in this way and you use the precomposed character, it turns out they're not really the same character. In Arabic the language, for instance, you would never put a hamza above a beh. You just don't do it. The beh doesn't get the hamza ever, so that wasn't a thing. But in this other language that uses it, and I can't even remember the name of it....

UNIDENTIFIED FEMALE: Foula.

ANDREW SULLIVAN: I'm sorry?

UNIDENTIFIED FEMALE: Foula.

ANDREW SULLIVAN: Ah, thank you. In Foula, you do and it's a different character and so you don't have this combined thing. And in fact, if you use the combining hamza above on a beh character that's an ordinary one and you use the combined one, they don't look exactly the same. To somebody who doesn't use Arabic normally, it's hard to tell the difference, but they don't look exactly the same.

An example of this in Latin actually is the "o" with the stroke that is used in certain Scandinavian languages (ø). Actually, you can produce that in Unicode with an "o" and a combining solidus that goes through, but it doesn't look the same. It doesn't look the same to even somebody who isn't a user of that character.

The final problem about this is "it doesn't look the same" is not actually the property that's defined in this. Unicode has nothing at all to say about how this looks. What it has to say is how you put the things together. But the characters that appear on your

screen or on your page are somebody else’s problem, which means that we can’t do this in terms of how it looks on the page or how it looks on the screen because there’s nowhere to put that. That’s really just a problem of your display or of your printer or something like that.

I know that sounds really crazy. Nobody gets to tell you what it looks like. But really, the font designer gets to decide how it looks, and that’s why all of your emojis, by the way, look different on Android and on Apple phones. That’s the same reason. That’s a font design problem.

This is an ongoing problem about the many layers of this. At bottom, the problem here is that writing systems are hard. Writing systems evolved over time and they evolved by communication with somebody else, which is why the Cyrillic A, the Latin A, and the Greek Alpha look kind of the same. They look kind of the same because they have a common ancestry, but they’re not the same letter. That’s really what the same problem is here, which is why you have a Cyrillic A and a Latin A even though you can’t tell the difference in capitals.

DON HOLLANDER: Blue chocolate or red chocolate? And come up to the microphone.

UNIDENTIFIED MALE: [inaudible], ICANN Coach, Deputy Manager of the [Marshall Islands] ccTLD. You are talking about variants. I think it is right that this issue could be handled by registry within the same language, but it is a problem for a registry to handle the variants for the same language. It must be handled by the same script to increase the reachability. Maybe I use character, for example, [Maca] Heh Marbuta. Maybe it is used by other language with another shape with different Unicode [inaudible]. This will be prevent Afghanistan or Pakistan people from accessing this website.

ANDREW SULLIVAN: Yes, but there are some obvious examples where that's going to be problematic. Every gTLD in theory has a global scope, .com is in theory for the whole world. So there's no way to know when you're looking at the name with the .com at the end what the language is supposed to be. In a lot of countries, of course, there's more than one language also, and so you have this problem.

I agree with you that there are a number of things that we can do, and one of the things does have to do with figuring out how to make sure that the repertoire of characters that you allow in a

given zone, how they relate to the scripts that are going to be used in there.

So you're exactly right about this, and there's a whole project here at ICANN, the Label Generation Rules and so on, that has to do with this and it's very valuable work. The work that has gone in there I think will inform the way we unlock the versions of Unicode, but it's still a combination of policies and protocols.

I agree with everything you've said, but I think the final thing that I want people to take away from here is there is no magic solution. There's no way to make this perfectly safe ever. There are always going to be some corner cases that are dangerous and scary.

When you go from 37 characters, which is the traditional LDH repertoire for labels in the domain name system for instance, when you go from that to all of the characters that are possibly valid for an IDN label – which is not the whole of Unicode but it's a subset of it – it's still a very large number of characters. When you do that, you increase the attack surface. That's the nature of having more options. When you have more options, you also have more potential for abuse.

There's a thing that we have to balance between these two things. When people say, "Don't worry. We're just going to lock it

down and it's going to be perfectly safe," that's the wrong way to think about it. What you have to do is risk-and-reward.

The Internet in some ways made us less safe. The Internet made us less safe in the sense that somebody can steal my credit card now, whereas in 1850 credit cards didn't exist and you couldn't steal it over the Internet. But the advantage is now I can go to Amazon and I can use my credit card. This is the sort of trade that we have to make.

So I think it's very important that registries do a good job of this, and I really encourage everybody who is involved with a registry here to make sure you have good policies because it's super important. Don't do things you don't understand because you're shooting your users in the foot. That's a very valuable point.

[PATRIK FALTSTROM]: [Patrik Faltstrom]. Let me continue on what Andrew just explained. I heard a comment that the registry policy can handle these kind of things. What we have to remember is that we are talking about the DNS protocol that you're using for every label on every level of the domain name.

You have ICANN and the root zone which has to have one set of rules. Then on the second level, you have potential ability to have one set of rules per registry. Then on the level below that,

you might have one set of rules per domain name you have down there. And then if you go down 15 levels, you might have I don't know how many different kind of rules.

But at each level, you have something that you can view as the registry. That's why ultimately it ends up being what Andrew points out. It ends up being the software, the editor you're using when your editing your zone file or your DNS software that compiles and actually eats/consumes whatever you have entered into it. That ends up being the policy, and that's where the policy of the registry meets the protocol definition. That's why we need to talk about these kind of things as being one thing and, as Andrew said, without context unfortunately.

So let me just emphasize as well being one of the persons that you should not give chocolate. You should probably beat me up as I wrote that stupid standard that we have problems with. I actually asked. I was late. I asked what was the difference between red and blue chocolate. I wanted to know whether one was poisonous or the other one forced me to tell the truth or something, or made me tell the truth.

To some degree for this specific thing with the combination of this character that may emphasize that to some degree there is no right answer but there's also no wrong answer. We just need to make a decision here, but we don't really know what to make

the decision based on because both solutions do have negative implications. We just have to make a choice.

That scenario planning is now moving forward, as Andrew said, partly thanks to what has been done with the label generation rules here in ICANN so I am also hopeful now after having this be installed for quite some time. Thank you.

BARRY LEIBA:

I wasn't going to mention this because it gets into the really geeky bit, but you mentioned variants and I wanted to point out that Arabic Beh With Hamza Above is not a variant and Latin A and Cyrillic A and Greek Alpha are not variants. There is a specific thing called "variants" in Unicode, and there you're talking about, say, Arabic Beh has at least four variants. There's one when it stands alone, one at the beginning of the word, one in the middle of the word, one at the end of the word, and they look slightly different. It's a real difficulty in sorting that out, and I know the Arabic registrars have all these algorithms for trying to fold the variants into one character and not issue variants and that's a true mess.

But what I wanted to say – and I'm going to go up here and turn around because I want to ask everybody a couple of questions – how many of you are here because you actively use internationalized domain names in some way? You don't count

but, okay, we have a couple. There are not too many. How many of you are here because you want to use them but can't effectively do so, who did not already have your hand up?

I'm trying to get a sense of why most of you are here. Are you here because you're interested but it's not of practical use to you right now? Can somebody just shout out why else you are here. What are you trying to get out of this?

UNIDENTIFIED MALE: gTLDs.

BARRY LEIBA: gTLDs.

UNIDENTIFIED MALE: I'm here because I'm working on one of the generation panels and have totally lost track of the fact that the average Internet user is average. He's not a brilliant person. He doesn't know all about the nitty-gritty details of how the Internet works, how the domain names work. So if you give him a .com in Latin and then you give him a .com in Cyrillic where the lowercase "m" looks like an uppercase "M" he has decades of experience that say uppercase and lowercase are interchangeable. Those are the same. But under the rules we're writing, those two Ms don't look

the same, so there's no bar to making those two TLDs which seems to me to be insane and I'm trying to figure out what the reasoning was.

BARRY LEIBA:

And one final thing before I sit down [providing we transfer the microphone] is, how many of you can now go back to some entity and encourage them to help universally accept this stuff? How many of you can actually take action to do that? I've got just one, and that's unfortunate. But anyway, I'm going to transfer the microphone and get my chocolate.

UNIDENTIFIED MALE:

Thank you. My question is to IETF. What about mixing between [inaudible] languages and [inaudible] languages within the same domain name or within the e-mail address? How could that software read this e-mail address or read this domain name?

ANDREW SULLIVAN:

There is a problem, a practical problem, with how this stuff works. The Internet, remember – this is another one of my soapboxes – but the Internet is not a noun like “the table.” It's a noun like “the traffic.” It's a mass noun.

That's because the Internet is a network of networks of networks, etc., all the way down. What that means is that it [distributes] administration. Lots of different people are administering these different things. This was precisely the point that [Patrik] was making about how there are registries at many different levels of registries. Everybody who runs a domain name like a zone is a registry in some sense because it runs the registry for that little zone, even the ones way out at the edge.

The effect of this then is that you can't central rules. That's why the rules for the root zone and the rules for the com zone and the rules for the CN zone and the rules for the CA zone are different and why the rules for anvilwalrusden.com, which is a zone that I operate, it's rules are different. My policy is names that I made up go in this zone and nobody else's names go in there.

So we have all of these different policies at every layer. Remember, each one of these people is a different authority. That's a really critical part of this. It's a distributed system. Each one of them is a different authority and nobody can make rules about the people below them because you can't enforce them. ICANN can enforce them because it has a contract. But ICANN can't enforce rules on me because it's hard to enforce rules on two things deep. And anyway, I'm also under a CA registry and

there's not contractual relationship between a country code and so on.

So you have all of these ways that these things interact with one another. That means that you can have different scripts at different places in the DNS tree, and there's nothing that anybody can do about that.

Now if you are operating a domain name, a particular domain name, it would be very unwise for you to mix this up too much because your users are going to be confused. If you want a positive experience for your users, you're going to avoid that. Similarly, it would be crazy for you to offer services in a language or a writing system that you don't understand because you're going to have a hard time supporting your customers in that way or you're going to have a hard time supporting your users.

But this goes back to that thing that I was saying earlier. That is, we want to align the incentive and the advantage. The things that you're going to promote are the things that are appropriate for your kind of use.

That was one of the ideas, going back to something I said earlier, about the way the e-mail internationalization worked. It assumed that users at either end of this connection were likely to be able to speak the same language. Why did it assume that? Well, because if you send me an e-mail in Farsi, I'm not going to

be able to do very much with that except delete it because I don't read Farsi.

That just means that we're going to have a natural path through this system. That means that I'm going to be able to read the e-mails in Farsi if I want to be able to talk to the person who is speaking Farsi, in Farsi, because if I can't, I can't communicate with them in the first place.

That's not necessarily true of domain names because I just click things on the Internet. But you don't just click an e-mail address. You have to compose the e-mail. That's maybe one of the differences between these two things.

The IETF, or for that matter anybody else, was not in a position to make universal rules about exactly how everybody has to do this because it's the Internet and because of that distributed operation we can't make a universal rule. All we can do is try to make things line up so that the incentives to do the right thing are there.

The final problem here, of course, is that people who are trying to commit fraud and so on, who are trying to do phishing and so on, they have an incentive to use this gap to their advantage. That's where these guidelines and so on are necessary because the guidelines allow you to do automatic checking of things so that user agents such as browsers could actually check to see

whether somebody is following a policy that they declare they're using or something like that.

That's a long-term vision, but we're not there yet.

DON HOLLANDER:

Thanks, Andrew. I actually don't know who you are, but I'll find out shortly, said he's here for the gTLDs. One of the things that UASG is also focused on is making sure that the new ACSII gTLDs work in all applications as well. And spoiler alert: they actually don't, so we're working on that.

BEN McILWAIN:

All right, yeah, I'm Ben McIlwain with Google. I just wanted to talk about gTLDs because we haven't really talked about them yet. We're launching a new one in May, .app. So we're obviously invested in them being as universally accepted as possible. But I think it's a commonality of purpose. If they help each other, internationalized strings and domain names and e-mail addresses are more likely to be accepted widely when people are already having to go into their systems and modify things to accept new TLDs and such.

So I think it's worthwhile to whatever extent possible help the other efforts out, like old-school regex validations of e-mails was literally a hardcoded set of a small number of TLDs. And it has

gotten better and our stuff at least has finally gotten to the point where it literally is there an @ sign and is there at least one dot in the domain? I guess you can have host names and so [you don't need] domain names, but there are very few restrictions and hopefully that applies globally to everything.

DON HOLLANDER:

Thanks very much. So you can choose red or blue. When we did this review of 1,000 website and how UA ready they were or weren't, we looked at the ones that were quite accepting and the ones that weren't quite accepting and regexes [work today]. I was surprised that in 2017 there were websites that still use hardcoded regex with specific ccTLDs or specific domain names. So we found one that accepted .com explicitly and accepted .CN explicitly but didn't accept many of the others. So for me, it was a bit of a shock.

We are looking for a universal regex that we can advocate and if you've got one, we would be keen for you to share that with the UA-discuss list and we can look to publish it.

BEN McILWAIN:

There is actually one that implements the full spec online. It's like 17 pages long or something. It turns out that regex is

actually not the right mechanism by which to verify this stuff anyway. It's possible, but it's horrific.

DON HOLLANDER: Yeah, we actually concur that we think they should be using a library call that is able to do these things better, but people seem to like regex.

Other comments I have read and....

UNIDENTIFIED FEMALE: Wait. We have a comment online.

DON HOLLANDER: Thanks.

UNIDENTIFIED FEMALE: This is from Edmon: “The discussion about the issues are useful and need to continue to be worked out” – sorry, my eyes are bad – “and others refer to not an issue would be solved. The salient point for UA, however is that there are not excuses to implement IDN EAI but rather that the implementer should consider that they implement the UA which unfortunately is the nature of things. The Internet changes and it is good practice to update the systems and also a good idea to design code that is adaptable to change. Go back to UA and I think that therefore

important to look at the issue is to discover and where your system needs to be changed.”

DON HOLLANDER:

Thanks, Edmon. So the UASG has four core talking points. One of them is that IT developers should keep to current standards and keep their code up to current standards. If they did that, then this issue wouldn't be an issue anymore. That's again part of our advocacy.

There was a hand. Elaine?

ELAINE PRUIS:

Thank you. I just want to reiterate several points different people have made. You had mentioned earlier that there aren't any sticks on the Internet. You have to use carrots. But in the case of a new gTLD, the operators are bound by contract to implement IDNA 2008 and then any new policies IETF comes up with.

As an applicant in 2012 and having had to write those applications and submit language tables, the discussions with my clients at that point were about, can we actually support our customers if we offer this language? So for several of the clients we limited it to languages that people in our company could speak – so French, Spanish, Portuguese – and we missed an

opportunity at that point to make a full offering because there just wasn't enough internal knowledge or grasp. Nobody was a linguist and nobody knew what this table meant.

Not only at that time at that company but having worked with several other registry operators, it's abundantly clear that whoever wrote the Verisign tables that was the groundwork for most of the applicants' [tables].

So I'm trying to bring this around to the fact that the UASG is doing a really great service to the community in creating this documentation that the average Internet user can't even begin to think about. So it's super important that we keep up this work, and hopefully the IETF can break this impasse so that more information can be produced.

DON HOLLANDER: Thanks very much. Other comments?

DANIEL DARDAILLER: Daniel Dardailler, W3C. I had a question related to I think that our issue is that are non-related to the internationalization, such as the length of the TLD and things like that, and then there is the Punycode mechanism, whatever it's called. I was wondering if someone solved the UA readiness problem by extending whatever lengths they authorized. Can people always use

Punycode to solve their issue, or is it a different problem? Trying to find ways between zero UA and full UA of using Punycode.

ANDREW SULLIVAN:

So Punycode, the way that we did this in the DNS is not beautiful. I think the technical term for this is it's a "filthy hack." And not only did we do a filthy hack, we did it twice. The basic idea is that you take a Unicode form, which in the most recent documents is called the U-label. That U-label has at least one non-ASCII character in it of some sort. Then it is transformed through an ASCII compatible encoding through an algorithm called Punycode.

What that does is it renders the Unicode code points in such a way that it's only letters, digits, and hyphens. Then you put on the front of that this prefix called xn--. That prefix says, "Warning: ASCII label for transformation ahead." Then when your application gets it, it's supposed to know, "Oh, I see this xn--. I have to do this transformation on this label."

You can see the very first problem here for a top-level domain is that on the wire – that is, in the DNS – the label is always more than four characters long. It's always xn--something, which automatically means that it's longer than .arpa which was the one big exception of the TLDs back in the day. It was either two or three or arpa. That was your regex.

So one of the ways that we ran into this – the other way we ran into it was just these long TLDs. I personally turned on .info back in 2001, so I’m intimately familiar with the “TLD is too long” problem. But then the other problem is, “Oh, you’ve got this xn-- something. It’s too long,” and so you fail on that.

Then if you don’t fail on that, then you fail on the next thing of, “Why do I have this xn-- here and I can’t do anything with it” and so on. So really, the internationalization case is two steps of the same set of problems.

Now one suggestion that various people have made is to try to come up with a new identifier system that sits on top of the DNS. That’s what gets exposed to humans, and we keep the old thing running underneath. It’s like it’s under the hood, so you just do an overlay. This is what geeks do when we run into trouble. We add another layer of indirection. That’s our goal. So that’s the idea here.

The only problem with this is for this to work, we have to get everybody on the Internet to upgrade to the new identifier system. That’s harder than reinstalling Windows on your laptop because the Internet isn’t under your control, so you can’t get everybody to do it at the same time.

That is fundamentally what all of these problems boil down to. It’s a group action problem. We have a sort of commons problem

here where lots [and lots of people] will have to do these things all at once.

So I agree with you. These issues are related. But I think they're related only in the sense that what we have is a distributed action problem. We have to get everybody to agree to this, and that's why I continue to be supportive of this steering group.

DON HOLLANDER:

I can tell you when we do our tests – and we do test a Punycode, [ASCII@Punycode.Punycode] – and we think that should just work and it doesn't. I don't understand that.

UNIDENTIFIED MALE:

Thank you. There's one addition I would like to [do what Andrew said] because I know that Daniel and a few others have been around for a while. One of the largest changes from IDNA 2003 to 2008 is that in IDNA 2008 you can translate or convert from a U-label to A-label and back. However, many times you won't because it's always a one-to-one mapping between the two. That was not the case for IDNA 2003. I think that is really important.

So one decision we made was to ensure that the Punycode is just an encoding of the Unicode string just like UTF-8 to UTF-16, etc. So it's equivalent with the encoding.

DON HOLLANDER: One of the other things that happened from IDNA 2003 to 2008 is you dropped the requirement for an open dot. So in our tests, we're seeing mixed results as a result. Some applications are transforming the open dot to a full stop as a label delimiter and others are saying, "What?"

ANDREW SULLIVAN: Well, let me say something about that. One of the things that was true about IDNA 2003 – and some registries are still using it – but one of the things that was true about it was that it was only kind of about labels. It was sort of about domain names, but it wasn't really. That was one of the mistakes. We learned that very quickly because you had that separator problem.

Now for those of you who are not already bored by the level of geekiness that we have gone to here, in the DNS the dots that you're familiar with when you look at a domain name don't appear. In fact, in the DNS it is possible to have a domain name "..". You can have a domain name that has a period in it. But of course, you never see one of those because they don't work.

The reason they don't work is because we're used to using in what's called the presentation format – the thing that humans see – we're used to using that dot as a separator. But because of

this, when IDNA 2003 attempted to use it as a delimiter, what that meant was mostly the protocol was about the individual labels and then every now and then you had this thing that didn't appear anywhere on the wire that you had to cope with.

When in IDNA 2008 we said, “No, no, no. We're going to have a one-to-one relationship between the U-label and the A-label,” that meant we couldn't do anything about the separators because we can't see them. They don't appear in the thing that comes in from the Internet. We can't see them at all, therefore we couldn't have a mapping of those things.

What you can do is the client can do the mapping, and the client should be doing the mapping. There is, in fact, advice to do that mapping. But you're right. Not every client does it, and that's a whole other set of problems that remains difficult for this.

I think as the user population gets larger, I think that some of those conventions will begin to reemerge, but I agree with you that so far we haven't seen that very effectively and it does create interoperation problems. There's no question about it.

UNIDENTIFIED FEMALE: We have a comment online. Edmon: “In terms of addressing UA by simply extending the length, I think it is not a good solution because the dynamic nature of the root zone, i.e., that the new

TLDs are added is not taken into account fully. And if you ask me, I think it could be worse because then you take away the heat to do the right update. But if you must, I would say go ahead. It's better than nothing.”

DON HOLLANDER:

I concur. We think the best option is to do it right the first time, but sometimes that's overwhelming. The UASG has five verbs in its focus: being able to accept, validate, store, process, and display them correctly. We see most of the problems – not all of the problems – but most of the problems in the first two.

Being able to accept the data and if your field length is not big you say, “Well, I've never seen a domain name with more than 50 characters,” then maybe that's what you set your input field to. But of course, the specifications say you can have 253? 255?

ANDREW SULLIVAN:

Well, 255, but there's a little thing at the end, so it's effectively 254. You don't really want to know why.

DON HOLLANDER:

They just need to read the specs and set their applications up appropriately. There are two components there. One is the total length of the domain name, and the other is the maximum

length of each label. Of course, with IDNs, the developers need to make sure that when it's converted to Punycode with an xn--, so there's at least four extra characters and sometimes more. So if developers read the specs which have been around for a while now, then we could have a mission accomplished party. But that hasn't happened yet.

Mark, what else is happening at Microsoft around UA? You told us what the driving force for you and how you get engineering resources happening. What else is happening?

MARK SVANCAREK:

I think it's just execution on the existing things. So you were saying about gTLDs. Actually, our UA work did start before the focus on internationalization. It started simply with the expansion of the gTLDs and talking to browser people and antispam people saying, "Do you realize that the way that you're updating the product month-to-month isn't going to work anymore? You can't just maintain these lists and hope that they'll stay stable for some period of time. You have to keep going to a definitive source." So that was one of the places we started.

Then just making an inventory throughout all the products and services, all the places where domain names or e-mail names exist, whether they're obvious places like in a CRM system the

contact information is an e-mail address or not obvious places like when I'm doing error logging I'm storing something. Making people realize that these things really do exist, just that level of awareness goes on and on and on.

I try to codify these things as much as possible within our engineering systems. As I mentioned, the reason that Active Directory supported EAI was because at a given time the server and tools team said, "Thou shalt support EAI, IDN, RESTful APIs, IPv6," things like that. And that was great when we sold only boxed products. When we started doing online services, those teams said, "We're going to make up our own things and our own rules," which mostly meant no rules. So retrofitting those old control systems, there's been a lot of resistance to it and so they have to approach from a different angle.

Some of the things that are happening right now are just recognition that we need to have systems in place so that we're not playing whack-a-mole every time a new service or product is introduced. There should be explicit controls that are enforced by the engineering system not to cause overhead on engineers – that's how it always feels at first – but so that there are some things you don't have to think about. You will be compliant by design. And that's a never-ending struggle, but we can improve it.

GG LEVINE: GG Levine. I represent the .pharmacy gTLD. You mentioned raising awareness, Mark. I'm wondering whose awareness are you raising about new gTLDs and what is ICANN doing and what can registry operators do to raise awareness of just ACSII gTLDs to have them be recognized by different service providers and so forth? It has been a hindrance of some of some of the new gTLDs. And especially with the impending new round of new gTLDs I would think that would be also a huge issue. So just wondering what are we doing to increase awareness and universal acceptance of just the new gTLDs.

MARK SVANCAREK: My comments a moment ago were specifically around the tiny engineering groups that we have at Microsoft. But I'm going to hand off the larger question to Don.

DON HOLLANDER: The Universal Acceptance Steering Group is focused on a specific technical issue that application developers and application providers we think are not aware of universal acceptance issues. So we're trying to the IT community so that as your domain name, your registry gets more and more popular

the systems that people try to use just work because they caught it before somebody complained about it.

So our focus is on the doers – so developers and system architects – and their directors, which are their CIOs and senior IT managers. We are not [about] raising awareness of new gTLDs. We’re not about raising awareness of the issue in what I will call mainstream media. Nothing we’re doing is secret, but our target audience is the people who can get their systems working before anybody complains.

The example that I often give is when I talk to a CIO, when their boss, their CEO, when his cousin gets one of these newfangled domain names and tries to use it in his system and it doesn’t work, we’re trying to make sure that the CIO recognizes that and is not embarrassed by that call from “the boss.”

So the goal is to get the doers and the directors aware of the issue before the complaints start happening.

Now one of the challenges we have is when we talk to CIOs, two things happen. One is, “What?” The second is, “I get it,” and it’s a very simple concept, and they get it very quickly. And then the third is, “Who cares?” because I’ve never heard this as being a problem ever before. It has never made it up through his help desk issue at all.

So I'm keen for you to promote .pharmacy or .whatever other gTLDs there are here. If you can actually create complaints within the customer systems, that's actually sort of good for me. But our focus is getting to the doers and the directors before, and we're in the process of doing that. We have some very good documentation. We're reaching out to CIOs and application development communities.

We're focusing, for the next six months anyway, on CIOs in government departments at a federal level, national level, state, local because we think government IT people have a purpose beyond profit. Whereas, if we go to a commercial operation, they'll say, "Yeah, I know, but I've never heard. I don't care." But government people have more of a mandate to serve their constituencies. And if their constituency wants to choose a .photography as their sense of identity, then they should be able to use it.

And your members, your registrants, no doubt they will be engaging with health departments and so forth. If they can introduce us to the CIOs in those public hospitals or public health departments, we'd be happy to get those names and we will reach out to them and raise awareness to them. So we'd be delighted to work cooperatively.

Yes?

UNIDENTIFIED MALE: [inaudible] question [inaudible]. Where are you seeing breakage where .pharmacy doesn't work for people?

UNIDENTIFIED FEMALE: A reminder on time.

GG LEVINE: Okay, we're seeing it in terms of signing up for services where you have to enter an e-mail address.

UNIDENTIFIED MALE: So it's the e-mail address, not the domain names themselves?

GG LEVINE: Correct.

UNIDENTIFIED MALE: That's what I thought.

DON HOLLANDER: Dennis has the last question, though I'm responding to that. Early on in our tenure, we did see domain names just not resolve in browsers. Browsers said, "I've never heard of this before, so

I’m going to treat it as a search term.” That was, in fact, I think our first bug report.

Dennis?

[DENNIS CHANG]:

I just wanted to point the UASG has a issue logging tool where you as a registry operator, or any user for that matter, can log issues with a certain website or application and ICANN’s support team will follow up to that organization and explain what the problem is and [try] to drive that up to resolution if that’s possible. So that [was a helpful] tool there.

DON HOLLANDER:

I have 90 seconds left? I have 30 seconds left?

UNIDENTIFIED FEMALE:

You’re five minutes past actually?

DON HOLLANDER:

Oh. So thank you to the panelists here and in virtual space, and thank you for people contributing and the discussion. It got a little bit geeky. From my perspective, that was fantastic because I might be old but I wasn’t in that space in the early days. So I always like to hear that history.

For those of you who are not subscribed to the universal acceptance working group, you can go to UASG.tech/subscribe and join the mailing list. If you don't want to join the mailing list, okay. There's heaps of good documentation. We produce a report prior to each ICANN meeting as to what's happened and what's happening.

So thank you all very much for coming, and thank you again to the panelists.

[END OF TRANSCRIPTION]