KOBE – Tech Day (1 of 3)
Monday, March 11, 2019 – 10:30 to 12:00 JST
ICANN64 | Kobe, Japan

EBERHARD LISSE: …chair of the technical working group that organizes Tech Day that we have on every ICANN meeting. I think it's now on the 38th or 36th or something, I don't remember exactly, we're doing this since Sao Paulo meeting at every ICANN meeting. I manage the ccTLD manager of .na, which is in Namibia in case you didn't know that. And we have got a nice agenda today as usual.

The first one, you can come to the rostrum already, will be Mark Svancarek from Microsoft who will talk about the effort to put IPv6 on their [big] campus. John Levine will then speak about two million registered IDNs and what he found when he researched this. Paul Hoffman will talk about DoH Resolvers.

And before lunch, Jothan Frakes will speak again about the public suffix list. For ccTLDs this is less important than for others, but we have a new ccTLD, .ss, and there are three country codes that may change. Indian Ocean territories may change. Swaziland may change. And Northern Macedonia will change to the Republic of Macedonia. So this impacts their ISO code and will then from that consequentially impact the ccTLD. It's another issue but if it does,

it impacts the public suffix list. So it's quite good for us to know where this is, what to do, who to contact.

Then we'll have a lunch, and afterwards we'll have the host presentation. What we always do is we offer the host or a local organization if the host is not well defined a presentation of their choice, usually also reflecting a little bit on their [setup]. But it's always interesting to hear what they're up to.

Tom Barrett will then revisit his presentation from the APTLD but will take out the introductory stuff so it's more technical. And something which I am quite interested in is to hear from Chuan Guo about Alibaba's cloud DNS practices. Alibaba is a huge company in China and where I am in Namibia almost unknown. But never mind, it's a huge organization so the scale of the operations and the DNS services is quite interesting, I'm sure.

Then Tim April Tim April will do the standing presentation for the SSAC. And I will come to this in a second when I'm done with the run through. Then [John] Levine will speak a bit about universal acceptance. And Jaap Akkerhuis will present some looking DNS research one of he universities has done. And then we hear about some IDN in Thailand from the Thai NIC.

We have been asked by the local government, by the Japanese government, at 14:46 to observe a moment of silence where I'm asked to read this particular statement. I have arranged with Tim

that at 14:46 we will rise and we will bring this up. We can read it in silence. That will take about a minute. I've spoken with the local organizers. They think this is a respectful way of doing it. If anybody has any objections to this, he is more than welcome to leave well beforehand. Okay, that said, I just wanted to make sure that everybody is aware of this. I personally am not religious, but we are guests here. We are doing this the way this is appropriate and this is in a respectful manner.

Mark Svancarek, we will present from the Adobe Connect thing hopefully for the last time. Next time we'll probably do it with Zoom. Here is the clicker.

MARK SVANCAREK: Thank you, Eberhard. Hi, everyone. I'm Mark Svancarek from Microsoft. You might know me from UASG or the infamous EPDP. But also inside of Microsoft, I'm sort of the IPv6 whip. So I make sure that customer requests for IPv6 are escalated properly and that teams are working together in order to have some sort of a reasonable IPv6 plan.

I am not a network architect. I know that there are network architects in the room. I'm sure that if you wanted to play stump the band with me, you would be able to stump me pretty quickly. After the presentation, I'd be happy to collect questions and take them back to my network architect if you're interested. She

decided not to come to Kobe. She is skiing in Whistler. It's a terrible choice that she's made, but there you go.

And finally, this is a talk about our corporate network, our corpnet. This is not a story about our cloud services. I can't really tell you anything about our cloud services that is not already public. So there is a lot of IPv6 related feature work that is happening right now. But if it's not yet announced, I can't talk about it and that's not really the reason for this presentation.

So as you can imagine, Microsoft has a pretty big campus and a pretty large network. We have four major regions with lots of smaller collections. We have smaller mini campuses or even individual buildings. in the Puget Sound centered on Redmond, that's the main campus. There are other places in North America, Silicon Valley, all over Europe and the Middle East, Asia Pacific. But we have just this one AS for the most part. So about almost 800 locations in total.

This is a combination of [on-premise] data centers and various services in Azure. So even though I'm not talking about our cloud services, we are a consumer of our own cloud services. So that's sort of considered the corpnet.

Branch offices with WAN connectivity. Internet peering is mostly through AS8075. We have – Mike was asking me a few minutes ago – over 100,000 employees, about 230,000 end users of the

network, 1,900 line of business apps that are managed through MSIT, the IT department. So we're going to come back to that because that's an important consideration for IPv6.

And there's about 1.2 million devices hitting the network. And of that 1.2, something like half of them are user-type devices – mobile devices, laptops, things of that sort – and the other half are servers or sensors or security cameras, badge readers, things like that. And so this was a consideration when deciding what our IPv6 pilot plan would be because if you're thinking about an intranet of things, replacing all the badge readers, security cameras, and stuff like that is kind of a distraction from what we really wanted to do. So ultimately that will have to be done, but that's not really what is happening in the pilot.

Here's an internal history. I won't spend much time on this. We started investigating IPv6 ISATAP around the turn of the millennium. In 2006 we started deploying it more within the network. Mostly this was focused on engineering groups who had a particular interest in it. So Windows engineering, the server teams, Skype, things like that. Skype at that time was IPv4 only, so maybe not a good example. Lync, the thing that became Skype for Business.

Around 2011 World IPv6 Day it became strategic. It became strategic not just because of the increased focus on it but because

we were running out of space as you can imagine. So we were moving the public space to Azure. We started rolling out dual stack, and we did not at that time have a user network that was delivering IPv6.

In 2016 we started rolling out IPv6 in earnest to users on the corpnet, wireless and wired. That includes the on-prem data networks. I apologize for the formatting error there. We have three IPv6 prefixes in corpnet. But as it says in the starburst there, there's still lots of little isolated IPv4-only networks throughout the campus.

What we see in corpnet is that about a third of it is IPv6, 66% of it is IPv4. Now this is what's based on Windows telemetry. Working with the Windows telemetry team, this is what we get. Clearly, that's not going to get all the devices. They're not going to see everything on Windows. So this is sort of an estimation. And 22% of the traffic coming into the corpnet is IPv6, so that's in line with what you would expect.

Now hopefully this isn't old hat to everyone here. Here are the reasons why we got interested in IPv6 in earnest. One of the things is that the Apple app store started requiring it. And you may have noticed that you can't get a Windows phone anymore. Maybe you've noticed. Maybe you never noticed the Windows phone. So we've really doubled down on Android and iOS, and

iOS app sore requires IPv6 which means that developers have to be able to test those features.

We've acquired Minecraft, GitHub, LinkedIn. LinkedIn is very much into IPv6. GitHub is very much no, no, no. Minecraft, actually I forgot to check on Minecraft where they're at, but LinkedIn is all in on IPv6.

But the main thing is that we're just simply running out of IPv4 space, and we think that our current allocation is going to run out in two to three years. I don't know if any of you know David Huberman who works in OCTO. He used work for Microsoft, and he was in charge of – among other things – acquiring IPv4 addresses. Over the few years that he worked for us, he spent about $50 million on IPv4 addresses. And that's at the prices that we had several years ago.

Another reason, of course, is just the complexity of having a dual stack. You have to do everything twice. And there's constant questioning too of, do we really have to do these two things? Can we just do the IPv4 right now? Why can't we do this? Can we have a special case? Blah, blah, blah. So you wind up with this massive complexity and people trying to cut corners all the time. So this is added incentive for us to move on.

Here's just a little anecdote. I can't actually read these numbers from here but if you can see them, you can see that the price of

IPv4 addresses has gone up very significantly just in the last couple of years. So the price for a /16 right now in 2019 could be as much as $1.2 million, and that isn't even with an expectation that those are clean addresses. So we heard an anecdote about some guy who was trying to get a visa and his mobile operator had acquired addresses from the government of Iran and found out that he couldn't get a visa because his IP was blacklisted. So these are the prices for addresses that you don't even know if they're clean.

So what did we learn? I think I'm getting a little ahead of myself. What we learned is based on a proof of concept that we decided to roll out. Knowing that there were a lot of devices on the wired network that were not user devices and that couldn't easily be replaced and could not opt in because they were not user devices, we decided to focus on a wireless proof of concept. In general, Microsoft is moving toward an IPv6-only wireless only network. That's the ultimate goal. So this was in line with where we wanted to go anyway.

So the things that you see are that NAT64 and DNS64 are essential. We've created a single IPv6-only SSID that you can opt into. We have a dual stacked VPN. Actually, our VPN doesn't do well on IPv6 at all. It's acquired through a partner. We've stopped trying to write our own VPNs, and so now we've partnered with somebody else and it doesn't yet do IPv6-only.

We started rolling out this corporate network in multiple locations. Right now as of last month, we're in 12 locations in U.S. and the Europe and Middle Eastern area. In fact, the building that I was in a few months ago was rolling it out just as I was leaving. So I never actually got a chance to use it, but I guess they're bringing it to my building soon. The goal is to get IPv6 enabled everywhere but really have IPv6-only every we can get away with it.

I've already mentioned the VPNs. The things the you find are that dual stack hides a lot of bugs. This should be obvious, but we're always surprised by it. You have something running in dual stack, and then you turn off IPv4 and you find all kinds of things. When we first did the trial in the first building, we discovered that no one but two of our infrastructure vendors had IPv6 bugs. So that was kind of strange that we would have not just one vendor but two vendors whose bugs had been masked by our dual stack before.

So that was fixed relatively fast, but that really goes to the point that you have to work with your vendors. You can't trust that what it says in their spec sheet is actually correct. And you also can't count on them to have a good process for quickly turning around firmware bugs or especially IPv6 firmware bugs. So you really have to have a good relationship with them and have a way to check when they do updates and stuff like that.

Another tricky thing was that Android requires RDNSS. We added this to Windows, I don't know, about a year ago. You can thank John Brzozowski for that. He was working at Comcast and he drove us to add that to Windows. Our networking equipment already supported it. So we have a lot of Android in the campus because, as I said, we don't make our own Windows phones anymore. And we also require them for the guest network, but that's really a different thing.

Now applications are the big unknown. As I said, we have 1,900 applications that we know of that are administered by the IT group. And so we're going through a similar process to what ICANN does on universal acceptance trying to figure out which of these we can get rid of, which can be updated, which we can get somebody to fix, which ones are through a third party and we can convince them to create a new version. And you never know what the state of those are going to be. But even our products, things that we actually sell to people, we will occasionally discover that people are using IPv4 literals. And so, of course, DNS64 is not going to help you with that.

And then last, of course, whenever anything goes wrong, there is always a kneejerk response and people say, well, let's just turn off IPv6 or let's make it not preferred or something like that. So the bugs never get fixed.

Here's some extra information that you can read on our efforts. There's the APNIC blog, and also there's an article on PacketPushers.

Are we taking questions? Yes? Okay, so that's it for the main presentation. Are there any questions?

Oh, just a few more details. Sorry. On this IPv6-only network, you have to opt in and we're giving away Starbucks cards and stuff like that to incentivize people to do it. We only have about 300 people who are opted in right now, and that's across all of those employees. Part of it is because you have to be in a certain building to do it. Others just don't really want to deal with it. At any given time, there's less than 50 people on the thing and yet that's still enough for us to get good information about issues, whether they're hardware issues, apps issues, or whatever.

EBERHARD LISSE: Any questions? We have two standing microphones.

MARK SVANCAREK: Oh, no, it's John.

[JOHN]: I'm some random guy you've never met. I'm wondering, do you have any idea what sort of bugs you've been running into that

need to be fixed? There's lots of room. We know DNS multicast doesn't work and changing packet sizes doesn't work. But I'm wondering with some real life experiments here, what actually broke?

MARK SVANCAREK: We should probably take some of that offline, but I know a lot of it is apps that don't use DNS.

[JOHN]: Really?

MARK SVANCAREK: Yeah.

[JOHN]: Really?

MARK SVANCAREK: They have just sloppy coding practices, yeah.

[JOHN]: Okay.

EBERHARD LISSE: More questions? We have enough time. We are not strapped for time. Take your pick. For the remote audience that is listening or watching, please introduce yourselves.

GAVIN BROWN: This is Gavin Brown from CentralNic. Thank you, Mark. This is a very interesting presentation. I have a question about DNSSEC which is another topic which is interesting to us in this group and in ICANN generally. Obviously, with DNS64 you have the issue of how you do DNSSEC validation when your DNS server is lying to its users about what resource records are present [in a name]. Was DNSSEC validation around before, and how are you planning on – if you are planning on enabling DNSSEC validation – how are you going to deal with the DNS64 [issue]?

MARK SVANCAREK: What I was told is that within the corpnet there's not a lot of DNSSEC at all and it's not really a big part of the pilot program at all.

GAVIN BROWN: Okay, so I guess one of the things it's probably worth pointing out is that DNS64 breaks DNSSEC. So if a zone is signed, it isn't dual stacked. And a DNS64 resolver, its answers will be considered to be bogus by [something] that does DNSSEC validation. It's

something you need to take into account when you're planning on doing an IPv6-only network if you also want DNSSEC.

MARK SVANCAREK: Yeah, we pretty much sidestepped that issue right now.

GAVIN BROWN: Okay, thank you.

MARK SVANCAREK: Warren.

WARREN KUMARI: Warren Kumari, Google. First off, thank you very much for doing this and even more so for actually talking about it. I think it's really useful and helpful. You mentioned VPN issues. Is that almost entirely because of NAT64 stuff or just the VPN client you're using doesn't really do IPv6 natively?

MARK SVANCAREK: I think it's just literally the client that we're using because they're working on fixing it. So I don't there's anything fundamental. It's just their implementation missed something.

EBERHARD LISSE: And, Warren, you're, of course, more than welcome to report on experiences at Google with the same issue.

JAY DALEY: On your previous slide, you mentioned expectations of IPv6 not meeting the reality. Can you go into that a little bit more please?

MARK SVANCAREK: I can't give a lot of detail on this, but there has been just a recurring problem where you read a spec sheet and it says it supports something, and then you find in actual practice that either it doesn't or you'll be working with the vendor on feature requests and they will regress something. So regressions are not an uncommon thing, of course, but the rate of IPv6 regressions really raises the question of, how well are you actually testing IPv6? If you fix something here and it regresses IPv6, I think that your test process must be broken. So working with your vendors on that to make sure that they have as part of their test suite that IPv6 is included in it is something that you really have to do. You have to be pretty proactive about it. Make them understand that this is a strategic important thing for you and that you won't be able to accept anything that hasn't been tested to the same level as other features.

EBERHARD LISSE: What number of devices that you're purchasing from Windows [inaudible] are we talking about?

MARK SVANCAREK: I don't know, actually. I think I actually should know that and I don't. Sorry.

EBERHARD LISSE: Because the bigger, the more leverage, isn't it?

MARK SVANCAREK: Certainly.

EBERHARD LISSE: I mean, if I get two devices, it's not very easy to get a vendor to put a fix in. If you've got 2 million or 2,000 or 200,000 they probably wake up when you phone them.

MARK SVANCAREK: Yeah, they're very large numbers and increasingly standardized. So even five years ago, the hardware that you'd see in our various data centers would vary from region to region because they would be rolled out at different times and there wasn't really an incentive to standardize them. And we've gone away from that. So now all the data centers are pretty much aligned in what the

hardware is and there's a commonality with what's happening in corpnet as well. So at one time we were really working against ourselves by not having the standardization. Meaning that bug fixes applied somewhere didn't necessarily benefit other things. And as you say, just the economic benefit of having fewer vendors delivering fewer products. That economy was not there necessarily.

EBERHARD LISSE: Vicky?

VICKY RISK: Vicky Risk from ISC. If you give us longer to think of questions, we'll come up with more and more questions.

MARK SVANCAREK: And we are out of time.

EBERHARD LISSE: No, we are not. You are not getting away that easy.

MARK SVANCAREK: Joking, joking.

VICKY RISK: To the point about IPv6 reality versus the claims, I'm wondering to what extent did you try to redesign the network to take advantage of new opportunities with IPv6, and in particular I'm thinking about decentralizing your address control using Slack, using router assigned addresses because that's an architectural shift.

MARK SVANCAREK: That's right, and actually I had an earlier version of this slide that mentioned Slack because that is part of our architectural design. So thank you for catching that. Again, I could get more details talking to the architect. I'm sorry that she's not here. But that is a good question. And we have applied that both within our cloud services and within our corpnet.

VICKY RISK: Is your sense that that's been a positive change?

MARK SVANCAREK: Yes.

VICKY RISK: Okay, thanks.

EBERHARD LISSE: I would rather like to have another question because I saw John Levine, the next speaker, go out to take a phone call. Okay, so thank you very much. Give him a big hand.

MARK SVANCAREK: Thank you, everyone.

PAUL HOFFMAN: Do you want me to go now?

EBERHARD LISSE: Well, if you can.

PAUL HOFFMAN: [inaudible] John and I earlier.

EBERHARD LISSE: Okay.

PAUL HOFFMAN: Whenever there's an opportunity to tease John about something, I think we should take it. Hi, as you can tell, I'm not John Levine, but we've worked together well in that past. So we – oh, John, do you want to go or do you want to let me go. I'll go. So pretend John and I are reversed.

I'm Paul Hoffman for those of you who don't know me. I work at ICANN in the office of the CTO. This is not an office of the CTO presentation. This is something that I'm doing myself. This is going to be a very quick introduction to DoH followed by a discussion of some new technology that is being proposed for it.

I think probably most people who have been following along with the DNS space know what DoH is, but I'm going to do just a slide on it. Some of you have heard about DNS-over-TLS which you just take DNS and you shove it over TLS. DoH is actually doing DNS over HTTPS. Meaning it's a full HTTP stack done where the traffic you get is actual HTTP traffic but it's doing DNS requests and responses. So DoH came after DoH.

One of the reasons it came up was that browsers actually know how to do HTTP requests. That's what they do all the time, so this was a little bit more natural for them. And it was also developed partially because, some of you may not know, but JavaScript applications running in a browser are very, very limited. They can't, for example, open up a port, but they can do random HTTP requests. So DoH would allow JavaScript applications to do real DNS. So instead of a JavaScript application being limited to give me the address of this domain name, they would actually be able to do all of DNS and do interesting things with that. But mostly this was because the browser vendors were like, "Hey, we do

HTTP. We know about HTTP caching, things like that. Why don't we just do our DNS this way?"

It was standardized last year, RFC 8484. Now I'm going to do a little bit of a warning here because many people are very aggrieved by some of the things that are happening in the DoH world. This discussion today is actually not about policy but if you want to hear more about DoH with policy issues, there's a meeting tomorrow on emerging identifiers technology where I will cover greater in depth about how DoH works but also will certainly talk more about the policy then. So not today please.

However, having said that, the thing that is bothering most people about how DoH has been implemented is how DoH servers are chosen. And that's because the standard actually didn't say how to choose a DoH server. Just in the same way that standards don't usually tell you anything about how you're going to do configuration, what's acceptable for configuring or whatever.

DoH didn't really talk about that. The RFC 8484 made some assumptions about how browser vendors would implement DoH and how they would allow you to choose a DoH server for your browser. But it turns out that those assumptions were wrong. That has caused a lot of grief for people because the way that it was wrong came out to be something where people have found

that problems that people would have objected to early weren't even brought up because of the assumptions.

So the main assumption we made was that your browser would show you a list of DoH servers that the browser vendor trusted. And that list might be 5, that list might be 50, but the browser vendor would have at least vetted a little bit because they don't want you going to a place that's going to obviously give you wrong answers. And that's not what has happened so far.

So right now, and I'll have another slide on this in a little bit, Firefox from Mozilla makes DoH visible in the normal UI. You don't have to do anything fancy. Chrome from Google does not yet make turning on DoH to be easy. And I've been told by some of the people on the Chrome team, it's like, "Hey, let all the Firefox people catch the grief on this. We'll figure this out." They haven't been pushing as hard. But the code is in there because it's very easy code.

One of the things that people, they look at a new protocol and they say, "Oh, look, all this work in the protocol." DoH, actually, the protocol part is extremely simple. It says if you have a DNS request that looks like this, here's how you turn it into an HTTP request, send it off, get an answer back, do the reverse and your done. It's really pretty minor. What's much more important is,

hey, you did that over HTTP. That's very different than the way the DNS people think. What are all those differences?

Now again, going to choosing a DoH server, where are you going to send these queries? Web applications – you know, JavaScript – they don't actually have a user interface. You've probably noticed that you never get popups from the zillion of JavaScript things that are running on any given page to ask you how to do things. So a web application that is also using DoH is just going to go to whichever DoH server it feels like. You're not going to have an option there, and people didn't really think that through.

The flipside of that is as far as we know not many web applications are using DoH yet today. But when that becomes more popular, the lack of user interface, the lack of choosability for people is going to become more significant.

So DoH got standardized and people said, okay, let's talk about this choosing part. And then they realized, we never really actually standardized a way of saying whichever server my capacity's already running, I want to DoH to there. I had mentioned it to people. They said, no, that's not really that important. People are going to want to do DoH to different vendors and such like that. So it didn't really get discussed.

Since we've been talking about this protocol, there's only one major browser that has it in it and this is the dialogue you would

get if you are running the latest release version of Firefox. You go into your network settings, which is hard enough to find anyway. It's at the very bottom of the General. At the very bottom of those settings you have these choices.

The most important thing to notice here is that list that I told you about has exactly one member in it, and it's chosen by default. Unless you're in the industry, you have no idea who cloudflare-dns.com is. So this turned from the user might make an informed to decision to, at least in the current version of Firefox, to the user has one decision that's if they turn on DoH is the one that they're most likely to use if they don't know to fill in the custom one. And they probably don't know who it is. Cloudflare is an infrastructure company. It's not a name like Google or Facebook or Alibaba or – and I'm sorry, I don't know all of the equivalents around the world of those – something that you would recognize and you might have an affiliation for or an affiliation against. Cloudflare is something that only the geeks know about, so that doesn't even really help them.

This list might expand. Mozilla has said that there's a program to allow more DoH servers to be in the list. But there has been no public announcement of how the list is being formed, what are the criteria for becoming a trusted resolver to be on the list. They keep saying that this will be coming, but it hasn't yet.

Now do note that there is the bottom choice there. So if you want to be running DoH or you're an administrator and you want your folks running DoH and you have a preferred DoH server such as you trust a certain open resolver that has a DoH interface or whatever, you can fill it in. But that's way beyond a normal user at this point.

Let's skip to what's happening now which is that there is a new proposal which has not been finished, and we'll cover that in the next slide, that allows you to find the DoH server that is associated with a resolver. So you open up your laptop. Your operating system chooses a resolver for you. You want a way to ask that resolver, I'd like to use DoH. If I wanted to use DoH, where would you send me? We don't have that now, and this is a proposal to add that.

That resolver might say, I do DoH. Send it to me. That's fine. But not all resolvers are going to do DoH. They might say, I don't do it but go over here because this is somebody who I trust as much as I trust me.

Given that, this is a proposal that's in front of the DoH working group in the IETF. It uses a special-use domain name where essentially you're going to ask the resolver a question to a name that no one else is going to be able to resolve. It's a name that's not going to be actually allocated in the real Internet. And then

you'll get your answers back from that. Or if your resolver actually also has a web server on it, you send it a well-known URI to the IP address of the resolver and you say, what do you want? And it will send back a list. The list might have zero saying I don't have a preferred DoH server. It might have one saying it's me. Or it might have a list and you pick. I'm being fairly vague here because this has just been started under serious discussion in the IETF within the last month.

So the next steps are, in fact, more discussion. This is the primary document being discussed in the DoH working group. Other people have been proposing more recently some policy documents. None of them have actually been adopted. At some point, maybe this year, it will finish. Trying to predict when IETF work will be done, an area director in the room just laughed. However, I am more optimistic than he is because I'm the document author. So I can help push things a little bit faster. However, he's possibly one of the people who will prevent it from moving forward, but we hope not.

Let's say it gets standardized. Then it has to appear in browsers. Fortunately, as you all have noticed, browsers get updated all the time, especially for features that they like. DoH came from the browser vendors initially, so I'm hoping that they will find this feature interesting and that they will wrap it into their DoH implementation and push it out soon after it's standardized.

After that, we still have the same problem – let me go back two slides – where user interfaces that look like that thing at the top users do not understand. So if we had another button up there that said your default as assigned by your operating system, that would be understandable to a small number of people but not everybody. How do we educate users if they want to be getting the protections of DoH to do that?

And again, this is completely irrespective of the larger policy issues of if you are going to someplace that you would not have gone anyway, what kind of DNS answers are you getting, things like that. And again, don't forget this is only really useful for browsers and web applications that choose to do it. If a browser doesn't want to offer this, you're still going to have the same policy issues. If you have a web application that doesn't want to do this, these are all things that can happen behind your back.

So that was a very brief overview because it's work that hasn't really gone on yet. And I'm happy to take questions.

EBERHARD LISSE: Anyone?

PAUL HOFFMAN: I'm especially happy to take questions from people who I don't know, but I'm okay to take them. I sort of wanted to give the other

folks a chance but if not, maybe they will cause you to want to ask questions.

EBERHARD LISSE: We have enough time.

WES HARDAKER: Wes Hardaker, USC/ISI. I think this is a good forum to discuss this mechanism in general because there are some policy issues surrounding the usage [inaudible] protocol that greatly affect how the domain names in the world propagate.

One of the things that always comes out of my mind is that this is one of the places that a lot of centralization is occurring because there's very few DoH providers and it is questionable in my mind that we're going to get a whole lot and that browsers are turning it on and selecting one by default.

And you talked about some other mechanisms for doing choosing and things like that, but it seems to me like we're heading toward a direction where if people ever end up in a network where they have to use DoH in order to tunnel DNS requests so that they can get the answers that they want, they will end up being on for every network at that point because it's not being picked on a per ISP kind of basis. Whereas, sometimes I may be able to use my local

resolver and other times I won't. So the end result is because it's unlikely every ISP will deploy this, we will….

PAUL HOFFMAN: Stop right there. I'm not sure that that assumption is at all true. Turning on a DoH server on a resolver is actually fairly trivial. I've been wrong about these predictions in the past, so I'm not saying you're wrong. I'm just saying I'm not convinced that it is true that ISPs won't turn it on. And we have large internationalized ISPs who have just put out a document saying we want to turn this on.

WES HARDAKER: Fair enough. And I think that it's easy to turn on, you're right. The question in my mind is are browsers and users going to be able to rotate easily between DoH servers, and that's what looks like that answer is quite possibly going to be no and single centralization is going to be more likely.

PAUL HOFFMAN: So, no, Wes. Let me just ask for a clarification here. I have a cable provider at my house. So I turn on my browser and it says to the operating system, which resolver are you using, and that resolver does have a DoH server. So that's of an ISP who I'm already using. And I travel and I keep that DoH server. So I'm no longer within that ISP zone. Would you call that centralization or not?

WES HARDAKER: That's a very good question.

PAUL HOFFMAN: It is, and it's sort of the crux of where I think you were getting at when you said I don't know if everyone is going to turn it on.

WES HARDAKER: My guess is that in that scenario you just described, the ISPs that are actually going to do that will be the large ones.

PAUL HOFFMAN: Absolutely.

WES HARDAKER: You just said cable provider, for example.

PAUL HOFFMAN: Yeah, and I agree. Large organizations are always more able to turn on new features. So does that concern you?

WES HARDAKER: It does because I think still the number of servers that are doing that are going to be small. But it remains to be seen. It's a concern I want to bring up so that I'm bringing education [inaudible]

room, not necessarily that it's a right or wrong at this point. It's a concern of the future.

PAUL HOFFMAN: It is a concern, and it's extremely hard to predict.

WARREN KUMARI: I've heard a number of concerns that browsers are going to do this and enable it for everybody and you will never be able to change it. I should point out that for a long time browsers have been able to do this sort of thing if they had chosen. As have all sorts of other apps. For example, Netflix seems to have been using its own resolver on apps for a long time built in their own. So I think that some of the browsers will do this and will force you to use it and you'll never be able to change it.

Maybe a slight kneejerk reaction to because it's HTTP and browsers know how to do that. However, that's unclear. I just wanted to mention that and keep it in mind that hopefully resolver implementation will make it really easy to enable a DoH server. Possibly in the future Unbound, BIND, etc., might ship with an easy way to enable it either in the software or with an add-on on the side. And then I think that pretty much anyone who runs a name server should also run a DoH server. If you're doing this, you can provide confidentiality, you can provide TLS-type protection.

So hopefully everybody deploys this and it doesn't end up a few small people running it.

And there's somebody behind me who might [punch me now].

PAUL HOFFMAN: But before you go, let me respond to that to one part that you said which was it's not just browsers, and that's very true. For all of us who have one of these in our pocket and who play a game on it, a game is essentially a narrow-use browser. Games know how to do URLs. It's all built in. Every game on your phone might be a DoH client. I don't want to go into that too heavily now because we're trying to get focused on the ones that are. But anything that is an application that can send URLs can do what we're talking about here.

WARREN KUMARI: So I have slight soapbox question….

PAUL HOFFMAN: Well, actually, now let the person behind you go.

WARREN KUMARI: Oh.

EBERHARD LISSE: It's getting a little bit too much debate. So we're not strapped for time.

[PETER HUDDLESTON]: Good morning, everyone. My name is [Peter Huddleston], the general manager of [CENTR]. I'm much more interested in the policy aspects of this which you are not discussing today as you pointed out.

WES HARDAKER: Right, but do please come tomorrow because that is going to be part of the [inaudible].

[PETER HUDDLESTON]: I will and thanks for that [pointer]. But these policy discussions, at least a large part, not all of them but a large part depends on some of the assumption that you mentioned earlier on. That is, for instance, the indication that Mozilla would be launching a program in which these resolvers could get certified or recognized or somehow appear in a dropdown list. That is not what I'm hearing from the Mozilla people in Brussels. I mean, they're the policy people, but they were pretty clear on that though. And the resolver of their choice would be hardcoded, baked into the product and that there might be even an option for the user to switch DoH on or off. But even that was up in the

air. And so I'm wondering at what point we're going to get those answers before we can then move to solving or at least discussing these policy issues.

PAUL HOFFMAN: Right, that's an excellent question.

[PETER HUDDLESTON]: And the other thing is Mozilla is less than 10% of the browser market. Google has 65%. Microsoft still has about 20%. And then the rest is less than 5%, Safari, Opera. So we're talking about five players that I think we're really desperately looking for an answer to before we can move into these policy questions that are not trivial, as you well know.

PAUL HOFFMAN: Right, and remember, the browser vendors change what they do all the time. So even if we get an answer soon, which I'm not betting on but if we do, that could change easily over the years.

[PETER HUDDLESTON]: A commitment from the browser vendors would be nice. A commitment from the resolvers too.

PAUL HOFFMAN: Long-term commitments from browser vendors would be nice. They're rare.

[PETER HUDDLESTON]: Yes. And a commitment from these resolvers too that they're going to respect some of the industry standards and practices which, being voluntary, is not a given. [inaudible] and things like that.

EBERHARD LISSE: I'm closing the questions after the last person standing there. And roughly two minutes per person I would propose.

RICHARD ROBERTO: Hi. Richard Roberto from Google. I have a question and I'm not sure if it should be asked today or tomorrow, so I'll ask it and you can let me know.

WES HARDAKER: Sure.

RICHARD ROBERTO: One of the reasons that I find DoH more compelling than TLS is simply because it's a better option for preventing censorship, and I'm wondering if the mechanisms discussed here actually don't

fully take that into consideration. Because if I have a DNS server that also happens to have a DoH server on it, it's just as easy for me to block that port on that server as it is to block the TLS port. So [inaudible].

WES HARDAKER: Yeah, so that is for tomorrow. But let me just for those of you who weren't following the question and I'll do [inaudible] tomorrow, but I'm not going to answer the question because it really is policy, many people wanted DoH because it allowed for a client who was getting blocked for policy reasons to get their DNS queries through to a place they trusted and get trusted answers. And that's a double-edged sword. You might want to do that because you don't like who's blocking. But you also might be relying on who's blocking because you have a legal requirement to. So all of this is very double-edged, but let's talk about that more tomorrow.

RICHARD ROBERTO: Okay, second question very quickly. I'm wondering, some of what's being discussed here seems to be more about applications policies rather than the DoH side of the equation.

WES HARDAKER: Yes, absolutely. Because again, every application that you've got on this guy has its own policies for who's it going to talk to and how.

DUANE WESSELS: Duane Wessels from Verisign. The DNS service discovery aspect of this makes me very nervous. I just browsed the draft quickly. I didn't see anything in there that would give me warm fuzzy feelings about preventing domain suffix searching and me seeing queries for whatever that name was .arpa.example.com. So is that something you've thought about yet?

WES HARDAKER: I've thought about it. If you have concerns, please send them to the list and they should be in the document.

DUANE WESSELS: Okay.

WES HARDAKER: Going back to the ICANN part of this, ICANN is fairly bottom-up. If you don't participate, things aren't going to change. The IETF is very much that way. That's how we got ICANN out of the IETF. If people like Duane have concerns about something, they need to

say it in the right forum so that the authors like me have to respond.

VITTORIO BERTOLA: Hi, I'm Vittorio Bertola from Open-Xchange, one of the other usual participants in these discussions, so we'll not reopen all the policy issues here. I went to the microphone because I was struck by the fact that you said that you're doing this as a personal contribution and not as a part of your ICANN job. Which is fine, but I think that at this point we're still waiting to understand what ICANN wants to do on this. Because the more and more we get into this discussion, the more we realize that it's more of a policy discussion maybe than a technical [inaudible] discussion.

WES HARDAKER: Absolutely.

VITTORIO BERTOLA: I mean, the technical [inaudible] is fine, but it serves [inaudible] policy objectives.

WES HARDAKER: It would be wonderful if the ICANN community would drive us on staff on what we should do about this. We haven't heard much. This is a start. There's going to be more tomorrow. But there are

policy aspects here, and ICANN staff should be driven by the ICANN community on policy, absolutely.

VITTORIO BERTOLA: Yeah. Even the questions that were just being done about promoting something that can bypass the censorship or however you want to call the fact that some governments [inaudible], this is not really technical. It's not even policy. Possibly it's politics. And it's something that [inaudible].

WES HARDAKER: Well, it's policy and there is technical aspects. Again, as you know because you've written some of these documents. There are very technical aspects to going around the resolver that your enterprise wanted you to be at. Your enterprise might be protecting you or they might be censoring. So going around, if they're censoring, gets you protection. Going around, if they're protecting you, is possibly a bad thing. So all of these things are very valuable policy things for the ICANN community to be talking about.

VITTORIO BERTOLA: Yeah, so the question was actually, how can – I mean, I've been trying to push some board members – but can we get some [inaudible]?

WES HARDAKER: Don't ask me. I'm on staff. You are part of the community. Please start the discussions. We will listen.

VITTORIO BERTOLA: Okay, thank you.

WES HARDAKER: Okay, thank you.

EBERHARD LISSE: All right, thank you very much. I found this quite interesting.

WES HARDAKER: Thank you.

EBERHARD LISSE: It's always good that we can start some things here. Give him a big hand.

And now since John Levine has finished his phone call, he can do his presentation.

JOHN LEVINE: As Paul commented to me privately, we turn out to be practically interchangeable anyway.

Hi. I'm John Levine. I am today talking about two million IDNs and what I found. I guess I should qualify. I always like to start by qualifying my audience. How many people here know what an IDN is? Everybody. Okay, that's great. How many people here could tell me the difference between an A-label and a U-label? Okay, that's about what I thought. That's fine.

So one day when I was supposed to be doing something else, I started grepping through my zone files. Because I have a subscription to the CZDS and with some degree of pain you can get copies the zone files for every ICANN contracted domain. So the number of zone files I have is 1,232 and the number of names in all of those zone files is 193 million, about half in .com and half in everything else. I wondered, "I wonder how many IDNs there are." Every IDN starts with xn—so that's easy to search for. I went through and it turns out there's only two million which on a computer is a pretty small number. So about 1% of the contracted names turn out to be IDNs.

So here in this chart that you can't read, I went through and just looked at how many IDNs there are in every zone. And the darker colored bars are the number of IDNs and the blue bars, most of which go off the top of the chart, are the sizes of the zones. What

this shows us here is that the IDNs are distributed very unevenly. All the way at the left is .com which is both by far the largest zone and has by far the largest number of IDNs. And then the next one, I believe, is .net and then there's .xyz which is very popular in China. And then you can see the dark bars get smaller and smaller. And then there are a number of fairly large zones that have some fraction of IDNs in them, but it's all over the place.

Here's what fraction of the names in a zone are IDNs, and it ranges from essentially 100% in some of the IDN zones down to nearly nothing. And you notice right in the middle of this chart there's a whole bunch of zones where exactly 50% of the names are IDNs? I was wondering what did that mean? It turns out there's a whole bunch of IDN zones that have been set up and are active but, in fact, have not yet registered anything. These zones each only have two names. One is the zone itself which is an IDN, and the other is nic.zone which is not an IDN. So this is simply an artifact of a whole bunch of zones that don't actually contain anything. But the realistic stuff is that the fraction varies a lot.

How did I do this? First I got all the zone files, which I already did. And then I wrote a Python script. And then I cranked everything through. And it turned out that I can go through every zone file in about 90 minutes which is good. The many times I find bugs in my script, I could fix the script and rerun it. And then I take everything out, and then it goes through and it tests every IDN and it also

puts all the statistics in a database. I also attempted for every IDN to find out when the names were registered since that is of some interest since the rules in the early 2000s were very different from the rules now.

That requires doing WHOIS queries. How many people here have ever done a WHOIS query? Yes. And how many people have ever done a WHOIS query that didn't actually work? Okay. So I went through and I got as many as I could. Also, the current redaction stuff doesn't affect this since nobody redacts the registration date. It's just the WHOIS is rate limited and it's very hard to get useful [answer out of those].

So I basically made three checks. There's the old IDN rules called IDNA 2003, and there's the new IDN rules called IDNA 2008. So for each name, I checked, is it valid under the old rules and is it valid under the new rules? Also, in all of the new TLDs and most of the legacy TLDs there are label generation rules. The TLD lists what scripts, what languages it will accept names in. And for every script, it actually has a table of the characters that are valid in that script.

So if a name is in Russian, the script is Cyrillic and all the characters have to be Cyrillic. And if the name is in Chinese, the script is Han and all the characters have to be in Han. What I did is I went through basically seeing to what degree do the names

match up with the label generation rules that are supposed to apply to those names.

So here's the theory of the label generation rules. It's in the contract. You can look at all of the TLD contracts and they say here's what languages we're going to accept. And then the TLDs are supposed to go and send the script rules to IANA in a standard format defined in RFC 3743. And IANA promptly publishes them on this webpage, the IDN tables webpage. So then all I have to do to find out what names are going to be valid is go and look at the IDN tables and everything will be great, right? Well, no.

The practice I have found is there are a bunch of new TLDs that have not bothered to update their contract and they register names in languages that are not in the contract. Which as far as I can tell is a total violation of their ICANN contract. It's one they could easily fix because amending the list of languages is easy, but they haven't. And not surprisingly, the most popular rogue language is Chinese.

The other thing I found is that nobody actually reads the specs. The table files sort of kind of follow RFC 3743 syntax. But each line of the file is supposed to have a bunch of Unicode code numbers separated by semicolons. How hard is it to type a semicolon? Well, for some people it's terribly hard. They use a dash or they use a slash or they use something else.

And some people even though these files are simple files of text, they turn them into HTML just because they could. So I then had to attempt to decode the HTML. And there are some TLDs that haven't even bothered to [sent]. And again, it's not clear to me whether this is a violation of the contract but, again, it's very unnecessary because these tables are available for every language you could possibly imagine, both from legacy TLDs and from the national TLDs. If you want a good Chinese table, you look at the one that .cn uses. If you want good Japanese tables, you look at the ones that .jp uses. So, again, they should, but they haven't.

So I wrote an ad hoc parser that I believe I managed to figure out all the ways that people managed to mis-enter stuff. What I did is each file simply has a long list of character codes and variants that don't matter here. So I turned it into in the Python language a set of here's all the character codes that are valid for this script. And then for each TLD, because a TLD like .com allows dozens of scripts, I merged all the sets together. So here is a giant set that contains every character that should be valid for any possible script for this TLD.

This gets pretty close to validating what script something is in. And I can check a name against the merge thing to ask, is it valid at all? And if it is, then I can go and check in the individual script. This gets pretty close. It turns out there's some rules that I'll

mention later that there's some characters that can only appear in certain contexts and this doesn't check the contexts.

So again, I'm checking whether they're valid under the old rules. The new rules, I checked whether they're valid under some set of characters. I checked whether they're valid for a specific language of script in the TLD. This isn't quite right, both because of the context rules and also some TLDs have updated their script rules. And for the ones that have done that, since getting the registration dates was hard, I didn't try to check the date. But it turns out the changes are pretty minor. So again, I think what I found is pretty close.

So the good news is what I found is the vast majority of IDNs are in fact valid under the rules that they're supposed to follow. They're valid under both the old IDN rules and the new IDN rules and the set of characters in each name is valid for some script defined for the TLD. But I found several thousand names that are not. I found 509 names that are bad under the old rules, and I found 4,800 that are arguably bad under the new rules.

For the ones that are [invalid] under the old rules, these are simply names that have been registered under the new rules. The biggest difference between the old rules and the new rules is one thing you'll appreciate is that ß is now a valid character. So if you speak German, this ß is a letter. In the old rules it wasn't a letter.

So now particularly in like .berlin and .hamburg, there are lots of names with that ß because that's how you write German.

Also, under the old rules you couldn't have an Arabic name that included digits because Arabic is written from right to left and digits are written from left to right. But it turns out there are conventional ways to combine them, and so the new rules do allow Arabic with digits. Anyway, all of these appear to be names that were registered recently. They follow the new rules, so it's not a problem unless, of course, you run a browser that follows the old rules like, say, Chrome.

For the new rules, I found there's a whole bunch of names that are arguably invalid under the new rules. About 1,000 of them are old names that are not valid under the new rules. All of those old names are junk as we'll see on the next slide. There's also an arguable issue about A-labels and U-labels. The name that's actually put into the DNS is called an A-label. It's ACSII xn--hardtoreadjunk. And then that's equivalent to some Unicode string in the actual language which might be Chinese. But it turns out that the encoding is of variable length. So if there's a lot of repeated characters in the Unicode, the U-label can be much longer than the A-label. And there are 3,000 names where the U-label is longer than 63 characters which is arguably invalid. The spec doesn't specifically outlaw it, but there's a phenomenal amount of code that assumes that each label in a domain name

is only 63 characters. So if you use these, your code will probably break.

So for the old names, they're names like §sex.com and €-bank.com and 1000°.com. Every name I looked at was parked or for sale. None of them were in active use. There are also a few that look evil. Like this xn--google-36d.com turns into g̱oogle.com. But if you look very closely at the bottom of the first g there's a little thing which is a diacritical mark that the IDN rules allow. So actually, this was registered in 2014 which seemed kind of recent. And I asked Verisign and they said, well, actually in 2014 that name was valid. Although it wouldn't be valid if they tried to register it now. And as far as I can tell, no one is using this particular name maliciously. They just did it to prove that you could.

However, I found lots of names, like 600 Chinese names in .CLUB which is not allowed Chinese names. Again, there's no reason they couldn't, but they don't. So they should get with [inaudible]. And there are also a few domains that simply don't follow the rules. For example, in .tokyo you can have Latin names written in ACSII text and you can have Chinese names written in Katakana, Hiragana, and Han. And there's this middle dot which is valid in Japanese scripts but not valid with Roman. Nonetheless, here we have taylor・swift.tokyo which points at a website that says something. That is harmless although I would presume they

would also allow like paypal・com.tokyo which could be potentially malicious. So again, I wish that the registries would follow their own rules.

And finally, we have names like this. I gather this Chinese character means top. So this is toptoptoptoptoptoptoptoptoptoptop.top. And if you look it up, if I do a WHOIS, it says it doesn't exist. It's reserved. Except if you also look it up in the DNS, it does exist. It has a webpage. So again, my understanding is if a name is reserved, it's not supposed to be in use. So they should make up their mind. Personally, if they want to have toptoptoptoptoptoptoptoptoptoptop.top, that's fine with me. But they should either use it or not use it.

So the summary is 99% of the names are good. Several thousand of them are messed up somehow. There's a long tail of old junk which I hope we can just let sleep. But there are definitely new TLDs that either don't understand or don't choose to follow the rules, and I hope that we can help them follow their rules.

That is it, and I presume I have a few minutes for questions.

EBERHARD LISSE: Yes, you have. One is, how many Sunday afternoons did you spend on this?

JOHN LEVINE: Three, I believe. Once I realized that I could run through everything in 90 minutes, writing and debugging the code was pretty quick. The hardest part was dealing with all of the badly punctuated script files.

EBERHARD LISSE: Okay, questions from the floor? Mark?

MARK SVANCAREK: When you were looking at things that don't follow the rules, how many of them were emoji domains?

JOHN LEVINE: I am pleased to report that I found no emoji at all in the contracted domains because IDNA – remember everything was valid either under 2003 or 2008. Actually, the first time I ran this, I found five names in .asia that were completely invalid under all the rules. But since I know people at .asia I said what's going on and they said, oh, whoops. Those were tests. They're gone now. So as far as I can tell, all of the emoji are in noncontracted domains where, of course, the issues are different.

MARK SVANCAREK: Thanks.

BARRY LEIBA: Thank you for doing this so we don't have to. On the mixture of Romaji and Kana, my understanding is that in Japanese writing they do mix Kanji, the Kanas, and Romaji all together. So that would mean that taylor・swift.tokyo ought to be legal in Japanese.

JOHN LEVINE: The dot has to be adjacent to a Japanese character.

BARRY LEIBA: Thank you.

JOHN LEVINE: It can't be between two Roman characters.

BARRY LEIBA: Okay, thank you.

JOHN LEVINE: Yeah, but it took me a minute to track that down and determine that it really was wrong.

PAUL HOFFMAN: So bringing this back to the ccTLDs, the ccNSO since that's part of our day here, there is a study group on emoji's use in ccTLDs

which is not what you covered. But something that you said at the end in answer to the first question brought this up where you said there are no emojis. That assumes that you know a definition of the word emoji means. I just want to be clear for the people who are going to be reading the report coming from the ccNSO work party that, in fact, defining the word emoji has now taken us almost a full page. So some people assume an emoji is this; some people assume an emoji is that. We should not be making assumptions about the definitions without saying whose definitions we are using.

JOHN LEVINE: In my case, I'm using the definition – they key difference between IDNA 2003 and 2008, which I don't have to explain to you….

PAUL HOFFMAN: Correct, since I wrote them.

JOHN LEVINE: Yes. Is 2003 basically said everything is okay except for these things we might rule out or [won't] translate. Whereas, 2008 said here's an actual specific list of what's valid. Okay, so under 2008….

PAUL HOFFMAN: All symbols are [inaudible].

JOHN LEVINE: Yeah, no symbols. So basically, 2008 has a very narrow definition of what's allowed and nothing that might be an emoji is included.

PAUL HOFFMAN: Correct.

JOHN LEVINE: There's a [guy] in the back who might have something to say about this.

PATRIK FALTSTROM: Yes, I need to correct. What you said is the result of what is important here, and this is very important just because of what's going on in the IETF at the moment. In IDNA 2008 it's the algorithm that is normative, not the tables or output of the algorithm. So IDNA 2008 does not say anything about code points. IDNA 2008 talks about the algorithm. And when applying those normative rules, then you get the code points that you were just talking about.

JOHN LEVINE: Yeah, when you apply the algorithm, you do end up with a specific set of these are the allowed characters.

PAUL HOFFMAN: Right, but what I was trying to say that you conflated was the idea of symbols and emojis. And as we're discovering in the ccNSO working group, that's not an easy thing to do.

JOHN LEVINE: Yeah, well, like these things here, the ~, these are all old symbols. These are not….

UNIDENTIFIED MALE: Current emojis.

JOHN LEVINE: These are not current emoji, yeah.

EBERHARD LISSE: Okay, I'll take these two questions, but two minutes each at the most please.

UNIDENTIFIED MALE: Quick statement more than question. With the .CLUB assessment, registry agreements get amendments allowing them to have

more services. So .CLUB is actually allowed to have Chinese. That's just one point to point out.

JOHN LEVINE: Okay.

UNIDENTIFIED MALE: Also, with the XML versions of the language table, that's in LGR format which is RFC 7940 and that's a requirement for people to post them in that way as well.

JOHN LEVINE: Well, I'm glad to hear that, although the tables, every IDNA table on the current IANA site is in 3743 format.

UNIDENTIFIED MALE: Yeah, so that's the confusing part because you can't pass [PDT] in some cases without supplying a 7940 table. So that's the transition process. [inaudible].

JOHN LEVINE: Yeah, I'm happy to parse either, just give me something I can use.

EBERHARD LISSE: The e-mail address of John is clickable in the agenda, so anybody wants to help him to refine his script by sending their own versions are more than welcome I think to contact him.

JOHN LEVINE: Sure. Yeah.

UNIDENTIFIED MALE: This is [inaudible] from [Taiwan]. Basically, for the IDN you were talking about one of the issues is the people that really follow IDN rules to [inaudible]. But I think there is another important issue, and hopefully we can have more information about it. It's how many people are really using the IDN in the Internet space. Because I saw that maybe some of the IDN is registered in the domain name but not quite much is used. If you go to the DNS checking the [inaudible], most of them, maybe 99.9% is still ACSII. Very little of the IDN. So here's the issue, why do people register IDN but are not going to use that? That's an interesting point. Do you have any idea about [that]?

JOHN LEVINE: Well, I mean, they register these too. I don't know to what extent. As I hardly need tell people here, there's a lot of speculation. People register names because they think they can sell them to somebody else. Also, I think there's a certain number that are

registered experimentally. It is my impression that the most interest in IDNs is probably in China. Although again, I realize in China everybody uses WeChat so they don't use domains at all.

UNIDENTIFIED MALE: Well, usually you don't need [inaudible].

JOHN LEVINE: I've also seen a fair amount of interest from Arabic-speaking countries, Saudi Arabia and like that. But the only languages I speak well are written in Roman letters, so I'm not the right person to ask about what people are actually using this for. But I agree. They don't seem to be used very much, and I couldn't tell you why.

UNIDENTIFIED MALE: Okay.

JOHN LEVINE: Okay, thank you.

EBERHARD LISSE: Okay, thank you very much. Give him a hand please. Now Jothan Frakes will talk a little bit about the public suffix list.

JOTHAN FRAKES: Hi, everyone. My name is Jothan Frakes. I'm the executive director of the Domain Name Association. We work to advocate commercial uses of growth innovation and give voice to an industry of domains.

But I'm here in a personal capacity. I've been a volunteer with the Mozilla community for a span of almost a couple decades. And I wanted to just take the opportunity and I'm grateful the ccTLDs to have me here today to share about and give updates on what the public suffix list is and how it can help you and help you have information about it so that you can have your entries and information updated in a way that best benefits you as you serve your communities of ccTLDs.

Could I just poll the room a little bit and ask, how many of you are familiar with what the public suffix list is and how to access it? Okay, roughly maybe 60% of you. So I'll just give a fast overview of it, but I included in the deck the information resources so that you'll be able to click and follow and find out more so that I can be very respectful of your time as we approach the lunch hour.

Essentially, the public suffix list was created by developers who sought to understand with a bit more elegance what a TLD is or isn't. I'm using the term TLD very loosely here. But how many of you as ccTLD administrators or operators not only register at the second level but also register where domains are available at the

third level under zones that you also operate? Many of you. I see many of you do. So those are very valid top-level domains, and they're operated by you as part of your stewardship and operation [and] the administration of your namespaces. And there's no reason to be prescriptive about that. There's flexibility in the DNS, and each of you have made determinations or worked in the administration of your TLDs to set those up. Fantastic.

Now when somebody who is a programmer who doesn't have the patience to participate in ICANN goes and tries to do development or work with things and figure out what's a TLD, it's best that they have some sophistication in their software to be able to understand that they need to treat co.uk as something that there would be registrations at the third level of and then also .uk as well.

So people can go and search for information and they can find things like the IANA list which is the alpha list of the top-level domains on the Internet. And they could stop there. But once they would do that, they would have no knowledge of the namespaces that are also available within those different administrative regions.

So a group of different developers across a variety of different companies developed over the course of time different directories of TLDs so that they could help and understand and

build programmatic features that would effectively operate and treat TLDs well.

If you notice, the example I give is just from the header of the public suffix list, but it goes into a little bit more depth. For example, for [inaudible] for .ac, in the IANA list there's just an entry for .ac. But in the list that exists within the public suffix list someone from the community found the directory on nic.ac, found the different subdomains, and submitted this to this repository. And over the course of maybe a dozen plus years this has grown quite a lot.

Many different companies, many different purposes have evolved over why somebody needs to have intelligence about what a top-level domain name is. There have been a variety of different individual efforts where people have made static lists of what a TLD effectively is. Over the course of time, many of them have fallen off and the public suffix list is what I would jokingly say is the least awful but most comprehensive list that's available.

It's used with browsers. There's quite a lot on this slide, so you may want to digest it afterwards. But essentially, it was originally designed so that cookies would be handled with security. So that when you use cookies for session ID or authentication, somebody would not be able to issue a cookie for co.uk and then be able to pull all kinds of different information beneath that in all the

subdomains. So you need something that says this is a barrier at which you would see normal registrations.

A variety of different uses have evolved over the course of time. How many of you use an Apple device? Just a show of hands. And how many of you use Google Chrome as a browser? These are fantastic things that you can use to look for things on the Internet. And in some cases, you're working with a tablet or a mobile device, so some of the browsers actually incorporate a search and a URL input as one input box. A lot of these programs use some form of logic to determine how can I quickly determine do I need to send this off into search or do I need to send this off to doing some form of resolution. Whether it's DoH or DNS, it's irrelevant. It needs to know that it's a domain name. So Chrome, Firefox, other browsers use some form of determination to see do I send this to DNS or do I send this to search.

And other uses have evolved over the course of time. There is WHOIS software. There is Creative Commons web crawlers. There's a variety of different uses that have evolved over the period of time, including software libraries that incorporate the public suffix list. There's a variety of languages I've listed here on the screen that when someone goes to do programmatic efforts where they're maybe processing forms. Maybe they're parsing mail server logs to look for what they're trying to evaluate traffic patterns. Maybe it's an antispam solution that needs to know

where should I accumulate all of this activity. With something like the public suffix list, you're able to identify specific areas of activity where registrations are available.

There are two sections of this file. There is an ICANN section which comes from what I would assert are official administrative directories. There's another section called a private section below where people can incorporate other subdomain uses. An example of this might be CentralNic's eu.com or uk.com. But there's a number of others such as GitHub, different hosting companies, and dynamic DNS providers.

We as the volunteers are not prescriptive about how this is used. We just try to facilitate and coordinate and keep these records updated so that the libraries that have been built upon it over the course of time have a good accurate representation of what's there. And I'm here today purely as a volunteer just to say that this is an important resource because it can really affect universal awareness, universal acceptance of your strings. It can affect how they behave. Not by design. We're not trying to design anything other than to help people trying to do things do it in a more organized way.

There are a few standards that have evolved after such as DMARC. There are others. Let's Encrypt, for example. You notice there's a trend with browsers that seek to force things to be HTTPS and

identify when they're not. And that has led to a real growth in people putting wildcard entries in if they offer subdomain hosting or things of that nature. So we've had an incredible increase in the amount of private section entries.

And we've undergone a Security and Stability Advisory review. SSAC70 governs the public suffix list or lists like it where companies have their own directories of what a TLD is.

We essentially extend the elegance of top-level domain awareness. And the reason I'm here today, I think I come maybe every three or four or five years and just check in with the ccTLD community kind of as a service as one of the volunteers to say have this on your radar. Know about it. It can be a very powerful and helpful tool to you to ensure that your namespaces operate in a very good way in a variety of different libraries and communicating within a lot of the standards that are in existence.

We have a very streamlined process. It's done through GitHub. You can download the repository. You can create a [fork]. I'm not going to go through how someone would necessarily interact with GitHub as a check-in/check-out repository, but the steps are pretty easy. I've included them on this slide here. Anyone is welcome to submit a patch.

There are some validations that we do for the ICANN section and other sections to ensure that we're, in fact, receiving updates and

requests from the actual party who is administratively responsible for that string. The most effective mechanism we've found is to have them add specific text records or other things within a zone file that we can validate.

We've spent quite a long time trying to validate when we receive requests that don't appear to come from an administrator of a TLD. The only mechanism aside from, gratefully I know many of you. I have the privilege of knowing many of you many years. But some of you I don't know. So often we send an e-mail to the IANA administrative contact which no doubt you receive a lot of e-mail to, and it may be an unfamiliar request for validation.

So if you do see something that comes from one of the Mozilla volunteers such as myself or I think the other two are Ryan Sleevi from Google or Simone Carletti from DNSimple. We're the three kind of keyholders of this, and we continue this service as a legacy. One of the original people who had done this by the name of Gervase Markham with Mozilla recently passed away and it was kind of a legacy that he had given to the community.

So if you see anything that requires validation or you would like to have questions perhaps on how to update your entries, I would invite you to look at the directory at publicsuffix.org, validate that your entries look accurate, and then there's a process here on the

slide which I'm glad to answer questions on to help you through that process.

That was easy. Are there any questions? I usually get a lot of people who are really angry. Like who are you guys? Why does this list exist? Ah, I spoke too soon.

WES HARDAKER: Who are you, and why does this list exist? Seriously, it does fall into my category list of things that I hate to love because I found it incredibly useful and at the same time I really wish it didn't have to exist and that we could fix this in better ways than trying to keep this list up-to-date.

I recently wrote a Python parser, a third one because I wanted the fastest one I could possibly do. And I think I've come up with the [third] one, a really fast one that I haven't actually published the code for yet and I need to do that. So one of the things I actually came up here to say was thank you for writing the test code directory that goes along with it that allowed me to validate that my code was not broken or more appropriately fix it after it was broken. But thanks for the work because I know it has to be a hard project and it's impossible to find them all, but it has enabled me to do some pretty cool research that I think will be a future Tech Day presentation as well. So I appreciate it. Thank you.

JOTHAN FRAKES: Yeah, thank you.

EBERHARD LISSE: Okay, I'll take all three questions, but one minute each only.

UNIDENTIFIED MALE: This is [inaudible] from [inaudible]. Thank you very much for maintaining this public suffix list. Thank you much because [inaudible] is one of the biggest customer of public suffix list. I also thanks to Gervase. He was the first maintainer of this public suffix list, and I miss him. So thank you.

UNIDENTIFIED MALE: Good morning, everyone. [inaudible] from .ae. Thank you very much for the interesting presentation. My question is, is this intended, for example, for domain names that are available for a certain group? Let's say, for example, a government entity that has a subdomain but it registers these names for various government entities, independent other entities. Is this supposed to also cover that scenario? Or it's only for those who can anyone can register which is public registration? Thank you.

JOTHAN FRAKES: Thank you. The way that it works is that it can affect the administrative horizons of cookies as they behave. So there are some government-only subdomains of ccTLDs that prefer that there is sharing of cookies and there are some that do not. So it's not a simple answer. But if it is, in fact, treated like a TLD even though it's a restricted TLD. You may want to review this and I'm glad to spend some time with you to help answer questions and sort that out.

We often get input from ccTLD administrators where we'll actually get conflicting requests. It's good to be able to sit and talk these through because they have some impacts.

And the reason this exists, I challenge you. Do you see a developer constituency here at ICANN? People who are developers? There's kind of a cool assumption, I don't know if it's an accurate assumption, that IETF equals all developers which I don't think it's a right assumption because something like this would not exist. And yet it does, and it didn't have any nexus with ICANN other than to be able to download an IANA list and have individual relationships with some ccTLDs.

So I have the privilege of knowing many of you and having the opportunity to overlap these worlds and help it be so that it can work a little bit better. So thank you very much for your time. I definitely appreciate it. If you see me in the halls, I'm glad to

answer questions on this. I tried to load as much up on the slide and also be very quick so I'm not in the way of anyone's lunches. So thank you all very much.

EBERHARD LISSE: Okay, we'll meet at 25 minutes past 1:00 so that we can start on time with the next presentation at half past 1:00.

[END OF TRANSCRIPTION]