ICANN81 | AGM – Joint Meeting: GAC and RSSAC
Sunday, November 10, 2024 – 9:00 to 10:00 TRT

JULIA CHARBELIN: Hello and welcome to the GAC joint meeting with the RSSAC. My name is Julia Charvolen from the ICANN GAC support team. Please note that this session is being recorded and is governed by the ICANN Expected Standards of Behavior and ICANN Community Anti-Harassment Policy. During this session, questions or comments will only be read aloud if submitted in the chat pod. Interpretation for this session will include all six UN languages and Portuguese. If you would like to speak during this session, please raise your hand in Zoom. Please state your name and the language you will speak, if other than English, and remember to speak at a reasonable pace. I will now hand the floor over to Nico Caballero, GAC Chair. Please.

NICO CABALLERO: Thank you very much, Julia. Welcome, everyone. As some of you have been informed through the daily newsletter and other channels of communication for ICANN 81 registrants, every year on November 10th, residents of the nation of Turkiye observe a minute of silence at 9:05 AM to commemorate the death of the country's historic great leader, Mustafa Kemal Atatürk, the founder of the Republic of Turkiye. In observation of this commemoration, we will be delaying a little bit the start of this session by a few minutes, more exactly five minutes, and we will begin at 9:10 AM local time. I appreciate that our speaker, RSSAC

Chair, Jeff Osborn, is joining us in this slight change in start time. Thank you, Jeff, and thank you, Rob, for that. So please feel free to exercise your own silent observation of this custom. I'm told we are likely to hear the audio signal of the moment shortly before the minute of commemoration takes place. So we will start at 9:10 so that people leaving the room can make it back conveniently. So with that, let me welcome again. It's 9:02 right now. So in about three minutes, we should hear the siren, or I don't know exactly how it works, but don't be surprised. Welcome, Jeff, welcome, Rob. Good morning.

JEFF OSBORN: I'm hoping that I'm going to get a signal from my esteemed colleague, Dr. Caballero, at the time to quit, so thank you for that. Good morning to all of you here in Istanbul, or good day, good afternoon, and good evening to those of you joining us remotely. My name is Jeff Osborn, and I am the Chair of RSSAC, the Root Server System Advisory Committee here at ICANN. In my day job, I'm the President of ISC, the developer of open source software for DNS and other internet protocols. I appreciate the opportunity to speak with you today, and I look forward to an ongoing discussion between the members of RSSAC and the GAC. I hope this is the beginning of a long-running dialogue rather than just a one-time introduction to the RSSAC, the Root Server System, and the Domain Name System itself. Based on feedback from the GAC, we will be presenting in two sections today. Part One describes the Root Server System Advisory Committee and how it works within the ICANN multistakeholder system, and Part Two, an explanation of the Domain Name System and the Root Server System. We hope this is

informative and educational, and I look forward to your questions at the end.

NICO CABALLERO: Thank you, everyone, for observing that three minutes of silence.

JEFF OSBORN: Part 1, the Root Server System Advisory Committee. RSSAC has a very narrow scope of activity and has a purpose shown here that the ICANN community has provided us with in their bylaws. On screen, you can see, the scope of activity is to advise the ICANN community and Board on matters relating to the operation, administration, security and integrity of the Internet's root server system.

In the next slide, we take that a little farther. Again, this is directly from the bylaws so I'd like to repeat it. The ICANN shall have the following responsibilities. First, to communicate on matters relating to the operation of the root servers and their multiple instances with the Internet technical community and the ICANN community. The RSSAC shall gather and articulate requirements to offer to those engaged in technical revision of the protocols and the best common practices relating to the operation of DNS servers.

Second, to communicate on matters relating to the administration of the root zone with those who have direct responsibility for that administration. These matters include the processes and procedures for the production of the root zone file. Thirdly, to engage in ongoing threat assessment and risk analysis of the root server system and

recommend any necessary audit activity to assess the current status of root servers and the root zone. Fourth, respond periodically to the Board on its activities, respond to requests for information or opinions from the Board, make policy recommendations to the ICANN community and the Board.

The RSSAC is made up of voting members approved by the root server operators and some non-voting liaisons. Each of the 12 operators appoints a primary and alternate member, and each operator in total has one vote. RSSAC receives non-voting liaisons from the Internet Assigned Numbers Authority, known as IANA, the Root Zone Maintainer, better known as the RZM, the Internet Architecture Board, known as the IAB, and the Security and Stability Advisory Committee, or SSAC. In addition, the RSSAC sends liaisons to the ICANN Board, the Customer Standing Committee (CSC), the Root Zone Evolution Review Committee (RZERC), the ICANN Nominating Committee (NomCom), and as needed on an ad hoc basis to various ICANN work groups.

In addition, there is an RSSAC Caucus, which is a wider group of subject matter experts and includes all RSSAC members. Membership is by application and reviewed by the RSSAC Caucus Membership Committee. Expertise and experience with DNS infrastructure is required, as this is a group of experts expected to perform meaningful work rather than a place to learn about the DNS. There's a mailing list membership available, which gets invitations to all our caucus meetings and the mailing list, but has more limited input.

It is important to recognize how RSSAC develops policy. The characteristics of that process are structured, open, and transparent.

We develop, express, and obtain feedback on policies and plans concerning the root server system. RSSAC policies are by nature recommended to restate the long-established normative framework used to operate the root server system and to suggest continued developments that grow from that existing framework.

It is important to recognize what RSSAC is not. RSSAC is not an enforcement agency. RSSAC is not a supervisory body. RSSAC is not a representative for the root server operators, and RSSAC is not a representative for the root server system.

As stated, the RSSAC regularly offers written advice on a number of topics, all available on the ICANN website. Here are some sample documents that give you an indication of the type of information and the recommendations that we provide. They fall into a number of different categories. For example, the history of the root server system is documented in RSSAC 002. This tracks the beginnings and the growth of the root server system over the decades. We also publish guidance on how to measure the operation of the root server system, as you can see here.

We have also adopted a number of recommendations concerning the continued evolution of the root server system governance. This also includes specific advice to the ICANN Board concerning implementation.

This concludes my introduction to RSSAC, and now I will move to discuss DNS and the role that the root server system plays in the operation of the DNS. To be clear, the goal of this presentation is to

increase your understanding of the root server system and the root server operators. We will explain the role and purpose of the DNS, the role of an address resolver, the role of an authoritative server, how, when, and why a resolver consults the root server system, and common misunderstandings about the root server system.

Now we get to the DNS, the Domain Name System. We will lay down a basic explanation of how DNS works that will allow us to introduce the root server system so you can see where it fits into the entire process. Introducing DNS. At its most basic, DNS uses human names to find computer addresses. Humans know domain names; that's easy, www.amazon.com. Computers need IP addresses. Those are harder, like 18.239.62.181. DNS is the way that we find out that www.amazon.com is actually at 18.239.62.181.

In the DNS, numbers can change while names stay the same. So connected devices need the DNS to find other connected devices. Originally we imagined this as a series of computers and servers; increasingly the greater numbers are of things like smartphones and smart devices. In summary, DNS questions ask about domain names. The answers are given in IP addresses.

The benefits of DNS, the obvious one that we can all explain is it's human-friendly. It's easier for us to remember a series of words than a series of numbers. That much is simple. But probably equally important operationally is that the service becomes very portable. You can apply a domain name to devices you want people to reach and then change the addresses wherever you need to. You can change the hardware, you can change the platform, you can change the location, the topology, or

just about anything else. Because as long as you have that unique name, the DNS will always get you to the new online home.

This is a huge distributed network that is remarkably easy to use. It has a flexible delegated database that is hundreds of millions of directories and it is arguably the world's largest distributed database. DNS in a nutshell. Obviously it's a little more complicated than that in practice. To begin with, devices get addresses from resolvers. There are millions of resolvers in the world and resolvers basically can find and read what we can think of as the internet's phone books. The internet phone books are authoritative servers that are held by the authoritative folks hanging on to their part of the internet. That listings are like a phone book, the zone content, address information, and phone numbers that you'd find in a phone book.

So for instance, this really comes down to asking questions as simple as, what is the number for www.amazon.com, and the answer, last time I looked it up, was 18.239.62.181. It's a simple single question with a simple single answer. It happens in milliseconds. And this happens, we estimate, 500 trillion times a day.

The resolvers get their addresses from authoritative servers. Remember in the last time we said the devices get their information from resolvers, the resolvers get them from authoritative servers. In most cases, the resolver remembers having asked an authoritative server, and it remembers the address for the domain in question. This is called caching. Most of the time, the answer is simply in memory. But sometimes the resolver needs to learn a new number or confirm an old number.

So here's where we're going to step through the layers of the DNS and learn how this works. Depending on how much information a resolver needs, it will either ask nobody. It will just check its own memory, and it already remembers the answer from the last time someone asked it. Or in case two, it will go to the domain name's authoritative server only. We were looking for, let's say, www.example.com, and we know where example.com is. We just ask them, hey, where is the www part?

In case three, we've lost more information than that, and we only know that it's somewhere in .com. That's the top-level domain, .com. We then ask the top-level domain's authoritative server, where is the domain name? In the fourth case, where we have just taken this resolver out of the box, or it's just woken up, or it's just had its circuits fried by a lightning storm, it has to know everything. It first needs to know where do I find the root server system so that I can find the TLD's authoritative server, so I can find the domain name server, so I can find my answer. Put it back into the cache, and hopefully I won't have to do this again.

Now we're going to walk you through what this process looks like inside the resolver. So, on the left is a gray box. That is the resolver. We've got a flow process in here of basically yes and no's, and we're going to see how does an answer start at the top. If you see on the top left, the question is asked, what is the IP address for www.example.com? And what we hope to come up with to the right of that is the answer. Here's the answer of the address returned to the incoming device.

So, in the simplest case, we ask, what is the IP address for www.example.com? We ask the resolver, does the resolver know the answer to that question? In the first case, and most common case, the

resolver says, yes, I do. I have that information, and I'll give it to you. So, on the right, we can see the frequency of this is very frequent, and as a matter of fact, it's routine. This is how more than 90% of resolutions happen. The resolver is asked, do you have this address? The resolver answers, yes, I do. I've had it in memory.

Less frequently, we have nothing in memory. So, the question starts, what is the IP address for www.example.com? The resolver is asked, do I know the answer yet? The resolver says, no, I don't know the answer. So, it's asked, well, do you at least know the resolver for example.com? And it does. So, it says, great, example.com authoritative server, help me know what www.example.com is. We go to the authoritative server, which knows the answer. It reports the answer. We put it in memory. Now, we ask, do we know the answer? Yes, we do, and we give the answer. It's just a simple one-step-up process.

Now, let's suppose we don't know the IP address for the authoritative server for example.com. All we know is example.com. Now, we ask, what is the IP address for www.example.com? Do I, the resolver, know the answer? No, I don't. Do I know where to find it for example.com? No, I don't. Can I find the server, the authoritative server for everything in .com? Yes, I actually can, and so I'm going to ask it, what is example.com? The .com authoritative server answers, here's the IP address for the authoritative server for example.com, returns it back to memory. Do I know the answer? No, because now I have to go back and find that to find www. First, I found example. Now, I've got to find www. I've got to build this thing.

I send the example to the authoritative server. It returns the answer. I put it in memory. Now, do I know the answer? Yes, I do, and again, all I've done is return the correct IP address for www.example.com, but based on how little the resolver knew when I first asked it, it has to go through more steps.

So, now let's involve for the first time the root server system. It is believed that one out of five or ten thousand times the resolver knows so little that you have to ask it all the way down to the level of the root server system. So, what is the IP address for www.example.com? Do I have the answer? No, it's not in the memory. Do I know how to get to example.com to ask? I don't even know that. Do I know how to get to .com? I don't even know that. I know absolutely nothing. This is a base cold resolver.

What it always knows is how to find the root servers. So, I send them the question, how do I find the IP address for .com? The root server system, in its sole task, hands out the address to the authoritative server for the TLDs, the top-level domains, and that is what makes it a root server. We report that IP address. It goes to memory, and now the machine can start asking these much more basic questions.

Do I know the address for example.com? No, not even close. Do I know the IP address for .com? Yes, you just learned it from the root server system. Great, now can I ask the authoritative server for .com what it knows? Yes, we'll go to that, and it'll tell us how to get to example.com. We will take that address, put it in memory. Do I know the entire answer yet? No, we just knew two of the words. So, do I know the authoritative address for the server for example.com? Yes, finally, with all these steps

I know it. I'll ask, how do I find www.example.com? The authoritative server knows the answer. It gives me the IP address. We put it in memory, and now when I ask the question, do I finally know the answer to the question, what is the IP address for www.example.com? The answer is yes, and here's the answer.

If you understood that clearly and could recite it back to me, you can get a well-paying job as a DNS expert.

NICO CABALLERO: One thing, Jeff, thank you for this very detailed explanation. This is normally called a DNS resolution, and as a matter of fact, there's a very good course in ICANN Learn that explains the whole thing in a somehow simpler way. But I just wanted to add, Jeff, that all this nightmare of different steps and everything gets resolved to caching in the end, right, which makes up 90% of the DNS. But you go ahead. You explain it way better than me.

JEFF OSBORN: The important thing to remember from that process is it's simple in almost every case. They are more and more unlikely as you go down the stack, and that's the ones that are more and more complicated are more and more unlikely. In the absolutely normal case, all that happens is, it's too many clicks to go back, but we start with the box that says, do you know the answer? Your memory says yes, and you get the answer. Vast majority of cases.

So here's a more summary description. There are those three levels of things we ask. The first one we're asking is we know what we think of as a domain name. We know example.com. What we're trying to do is find the thing to the left of it. Is this www.example.com or mail.example.com or nameserver.example.com or one of those other things. There are 350 million places where that can be because there are roughly 350 million of what you think of as domain names that are out there in the world.

Top-level domain names are a much smaller group. If you have to go out to, for a nation, go to .uk and ask what's everything you know about .uk, or what's everything you know about .edu, or what's everything you know about any of the other ones of those, it's a much smaller list. There are about 1,450 of those entities in the world, and they tend to have between 1,000 and 10 million things within them. Those are maintained entirely by the top-level domain registry. So the people who own .edu and the people who manage .eu and the people who manage .uk are fully responsible for all of those names.

The root zone, in the grand scheme of things, is really tiny. It's one document. It's one zone. It's basically a list of those 1,450 top-level domains and an address to get to them. That's it. It is maintained not by the root server system, but by the IANA, and then it is cryptographically locked up by a group called the Root Zone Maintainer. Only then is it released to the public, and it is available for delivery and available pickup from the root server system.

In review, the root server holds a copy of the root zone. That's it. It's all the information we have. The root zone holds the addresses of the 1,450 top-level domains that are out there, and you know the top-level

domains. I like to think of it as the thing to the right of the dot: .com, .nl, .jobs, .cz—a whole bunch of things.

The TLD, each of those TLDs has an authoritative server that knows the address of all the next steps. So all of the names that end in .com, you can look them up at the .com authoritative server and find amazon.com, tiktok.com, freshflowers.com, asunnydayatthepark.com. Those are all known by the authoritative server for that top-level domain, and the same for the other ones I've included, .nl for Netherlands, .jobs as a gTLD.

And then when you get to the actual domain name, for instance, jeffsgarden.org could be an organization. I am the one who is responsible for all of the domain name additions to the left of there. The domain name's authoritative servers are controlled by the registrant who is the operator of the domain name. So there are fully hundreds of millions of authoritative servers for the domain names out there.

So you can see this is a hierarchy. The resolver simply finds and returns the answer. We threatened to put this on a T-shirt, so I put it in big writing. In the millisecond world of a resolver, queries to the root server system are rare. That's a really important outcome from here. Most things come out of memory, most things people already know. It is a rare instance when you have to go all the way to the root server system to look something up.

So about the root server system, it provides address information, not content. In fact, it provides a partial address. The root server system literally answers one small part of an address question. All that you're

asking, all the resolver is asking a root server system is, can you give me the address of where I can look up the addresses of the top-level domains? It's a part of an address; it's not even an entire address.

It is not content. The root server system does not host web or email or any other similar internet content in any other way. It doesn't deliver it, it doesn't transmit it, it has nothing to do with content. We simply serve up a small fraction of an address. So it's important to remind you, the root server system does not manage or carry internet content.

Furthermore, the root server system is not a gatekeeper to the internet. Root server systems answer queries posed by address resolvers, and address resolvers usually construct their answers from cache memory. They usually don't need to ask anybody else at all, but they have a series of other places to look before they get to asking the root server, and when they do get there, it's only because they had a really hard question and they're in a very funny state, having just come out of the box or lived through a lightning strike or something similar to that.

As a result, traffic is almost always transmitted without the need to wait on a root server system. As a good estimate, it's less than one out of five or ten thousand times any of this happens that involves the root server system.

The root server system is remarkably stable, secure, and resilient for a number of reasons. For one thing, it is massively redundant. There are currently just over 1,800 globally distributed server instances around the world. Each server has entirely 100% of all of the information it needs. You don't need to be able to get to a few of these to get your

answer. Any one of them can provide you the addresses of every top-level domain authoritative server.

We run on diverse hardware platforms. There is a great variety of hardware that's out there in use. We operate on a diverse number of operating systems. We operate on a diverse number of DNS applications on top of those operating systems, on top of those hardware systems. And we practice diverse routing so that where this information goes is different in every case.

What this means from a technology point of view is there is no single point of technological failure. A Microsoft bug would harm a tiny fraction of us. A bug in just about anything out there would harm a tiny fraction of us. This is something where we think it's vitally important to recognize the diversity of the system is its strength.

Similarly, the root server system is operated by 12 autonomous root server operators. Each root server operator is entirely independent of the other. The root server operators cooperate continuously with others. We all tend to know each other. We speak frequently. We meet frequently. But we're independent. We're independent in how we work, how we put these systems together, and what we do.

A result of this that's very important is that there is no imaginable force majeure event suffered by one operator, like a court injunction, as unfortunately happened with AFRINIC, that would have operational effect on any of the others. So it's important to recognize in the root server system there is no single point of institutional failure.

Next, the stability and security and resilience is something we really talk about a lot. And it's because we care about it a lot. We implement it. We work on it. And it's true. The system has operated since the 1980s, and there has not been a service blackout of any sort in all those decades. It's longer than some of you have been alive.

DDoS attackers have tried and they have failed by design. This is a remarkably resilient system on purpose. As a result, we have 30-plus years of operation, 7 days a week, 24 hours a day, 365 days a year. I'm repeating myself, but I think it's really important to recognize.

This is back to what it is that we do not do. The root server operators do not decide what is in the root zone. We are simply a reliable, authenticated delivery method. The registrant, the domain name holder, decides address information for their own domain. And they provide that authoritative server address to the TLD registry. The TLD registry decides its authoritative server address and provides that authoritative server address to the IANA. The IANA authenticates revisions to TLD authoritative server addresses and provides them to the Root Zone Maintainer. The Root Zone Maintainer then cryptographically signs the root zone and provides the signed root zone to the root server operators of the world.

So this information comes from the users through the TLDs, through the IANA, and to the RZM. And by the time we get it, it has been confirmed many times over and cryptographically signed. Besides us, it's freely available to the world. If anybody wants to check whether the document we've got is the same as the one you've got, it is trivially easy to check.

In summary, the root server system provides TLD address information, not internet content. The root server system does not decide what TLD information appears in the root zone. TLD information is provided by the TLDs, the IANA, and the Root Zone Maintainer. The root server system is not a gatekeeper. The end user has very little, if any, direct interaction with end users. There is nearly a zero likelihood that any of you have ever experienced a delay in the performance of the internet due to a root server-induced delay. It is extremely unlikely.

In summary, the system is resilient. It has no single point of technological failure. It has no single point of institutional failure. And I thank you very much for your time.

NICO CABALLERO: Thank you. Thank you so very much for this very detailed presentation. I know this is DNS 101 for you, but we needed to give the gist of it to the full GAC. Again, there's a very good—actually, I think it's called DNS 101 on ICANN Learn. Not sure about that, but that's probably the case. I strongly recommend taking, I don't know, maybe 30 minutes, I don't know, one hour, you know, in order to get—I don't want you to become DNS experts or software engineers or network engineers or anything like that, but it helps a lot understand, you know, the nuances whenever you're dealing with anything, as a matter of fact, anything related to DNS, like, I don't know, cryptography, for example. There's symmetric and asymmetric cryptography. I mean, we could talk about that for hours, but for the time being, thank you again, Jeff and Rob.

Do we have questions or comments or anything you would like to mention at this point for Rob or for Jeff? Go ahead, please, go ahead.

MARCO HOGEWONING: Thank you, Jeff, for a remarkably easy-to-understand presentation, and I hope this is—and there's never a simple answer, but I hope you have one. As one of my colleagues just whispered, I hardly ever type www anymore, and indeed, correct, if I tell my browser to go to ICANN.org, it will simply present me with the ICANN homepage. How does this affect the DNS resolver sequence, or does it affect the DNS resolver sequence that you just explained?

JEFF OSBORN: Marco, that's an excellent question. The important part to remember is that the words can stay the same while the numbers change. When you're clicking on an icon or clicking on an underline, it is still using the domain name words to say where it wants to go. Your application doesn't need to know that Amazon did a maintenance refit and moved its authoritative server to a different address because the phrase that you're using still will map to the correct address. They can take care of moving those things in the authoritative bit.

I explained this presentation to my son, who's 24, and he was trying to tell me how many years it had been since he typed something into a browser. So I had the same discussion. But again, this is the ability to map resources in a common way. You could use anything, and it doesn't have to be as human-understandable anymore. It simply has to be known that as long as you're calling it some string of characters, the

authoritative server knows to map it to the correct address. Does that help?

ROB CAROLINA: I'm going to do the world's most dangerous thing, and that is, as a lawyer, I'm going to try to explain something in DNS. So watch me fall. My friends will correct me. I believe it is the case that the question of whether or not something like ICANN.org resolves to www.icann.org is also something that is configurable by the registrant in that domain name's zone. So that's actually a feature of DNS that can be set and controlled by the registrant. Some registrants choose to default their domain name always to a given location. Others will leave it undefined, at which point you get a response that says, I don't understand what you're asking for. But these days, most commercial sites will default so that the www is invisible or implied. But that is a function of what the registrant has decided.

NICO CABALLERO: Thank you, Rob. Thank you, Netherlands, for the question. I have Papua New Guinea.

RUSSELL WORUBA: Hola, Chair, and thank you, our esteemed colleagues, for this presentation. Just a leading question towards the generic top-level domains that are coming in this round. How do the root servers weigh in that sequence? Does having a new generic top-level domain fit in? And if it matters at all, how it will be resolved?

JEFF OSBORN: Thank you for the question. The new top-level domains simply add themselves to the zone file, and instead of having 1,450 domains, there will be some larger number. What's interesting historically was, until about 20 years ago, the number of domain servers needed to increase for a technological reason because it was required to cover all of the addressing. Twenty years ago, an interesting technology called Anycast was implemented, which suddenly made it tremendously easier to cover very large amounts of information over very large areas very reliably.

And since then, there hasn't been an addition of a root server because we realized just how over-provisioned we were. So similarly, if we doubled or tripled the size of the root zone file, we don't see any reason to need to change anything other than literally adding to the name of the root zone file. If you think about it, a document with 1,400 lines of text is not a very large document. We all take pictures on our phones every day that are thousands of times the size of these things. So going from 1,400 to 3,000 or even 4,000 or 5,000 should have just about zero effect on how the root server system operates. Does that help?

RUSSELL WORUBA: Thank you.

NICO CABALLERO: I have China and the USA. China, go ahead, please.

WANG LANG: Thank you, Nico. Thank you, Jeff, for your detailed information. As you mentioned, there are thousands of instances of the root server system based on the Anycast technology and the Border Gateway Protocols. Do you think these instances would bring more risks or threats? For example, fraud or hijacking of authority.

JEFF OSBORN: The existence of fraud in the root zones is something that in 30 years we've never experienced anyone trying to have, but of course in the security world you always look for the difficult and the unusual and the rare. So I'm tempted, because I'm sitting with much better experts than me in front, that perhaps one of them might give an answer that is more thorough. So if you don't mind, I may defer this to my colleague from Sweden, Liman.

LARS-JOHAN LIMAN: Thank you. I'm sorry I sit with my back towards all of you. My name is Lars-Johan Liman. I work for Netnod, operators of one of the root server clusters. The fraud thing is something that has—we've tried to address, not we, we being the entire technical community on the internet, not we as root server operators. We've tried to address this with adding security features to the DNS data, the DNSSEC functions, secure DNS, where all data in the root zone and further down in the tree for those who choose to do so, but certainly for the root zone, all data is cryptographically signed, so it's rather easy to verify that the data you receive is correct.

And this is done on a local level where in the resolver that Jeff described earlier, so this can be done very close to the user. So the fact that this is distributed over many instances is not a big problem thanks to DNSSEC. It's possible to validate the information to make sure it's correct when you receive it very close to the end user. Thanks.

NICO CABALLERO: Thank you, China, for the question. Thank you, Jeff, for the detailed answer. I have the USA.

[LEN HAWES:] Good morning. Thank you, Jeff. I'm [Len Hawes] with the United States Department of State. My question is along the lines of the redundancies that were previously mentioned, but the question I have is, is there an ongoing process to evaluate, because the internet structure is constantly checking either the utilization rate or the technological diversity of those instances on an ongoing basis to establish that you've got a truly diversified system?

JEFF OSBORN: Excellent question. Thank you, sir. The interesting part of the root server system, as I said, is we really had a bit of a leap when we figured out Anycast. And if you'd like to get a boring, informative 10 minutes, corner me sometime in the hall and I'll tell you how that works. But the important part is we suddenly became over-provisioned and overly able to handle the volume. Much of the internet is involved with trying to figure out how we're going to address the increasing volume. We're

one of the few organizations where we've pretty much got it, so what we try to work on is ensuring diversity, ensuring if two companies buy each other we aren't suddenly not diverse because we now both use their hardware, ensuring that when there are forks in the operating system that we're all on enough different ones.

The other piece is for having enough units in place. For my own organization, I'm the president of ISC, we have several hundred instances in place and it is our program that allows if you wrote me an email today or asked me in the hall today, could we set up an instance in our own place? Within some pretty easy limitations, it would be functioning within, I don't know, depending on tariffs and shipping and things, 30 days, 60 days.

So we are wide open, there are other organizations as willing as us to just simply add new instances when required. We spend a lot of time also ensuring that, again, it's hard not to get technical, but topology and geography are very different and computer topology often means you could look at a geographic map and feel something wasn't working where topologically the computers were actually doing a very good job. We look at that pretty obsessively. So it's something we spend a lot of time on and we're also very open to anybody's input on how we could do it better.

ROB CAROLINA: To address specifically your question about assuring diversity, the root server operators in addition to participating actively in RSSAC also participate in other forums, which are—another forum in particular, not

an ICANN forum, which is focused on operational matters dealing with the root server system. That is, and the discussions there are not published because they get into very significant operational detail.

So to address your specific question, that group has been actively studying and surveying the question of infrastructure diversity amongst the root server operators because we've been asking ourselves the same questions for many years. And I can, although we will not be publishing the data for obvious reasons, I can assure you that we are very pleased to discover from our internal surveys that we have exactly the kind of diversity that Jeff described earlier.

NICO CABALLERO: Thank you, Rob, for the answer. I have the European Commission and the gentleman over there. Go ahead, please.

EUROPEAN COMMISSION: Thank you. And thank you for the very interesting presentation. I think it achieved its objective of increasing the understanding of the root server system. Just to say that we are following with great interest the work of RSSAC. Also we understand that you have been working on updating the governance framework and this is reflected to a lot of advisory documents that you have also recently updated. Just wondering—and I understand this is an ongoing process—do you have any latest updates or timeline how this work is currently going? Anything you can share with us? Thank you.

JEFF OSBORN: With all due respect, the RSSAC is not the GWG. There is an organization that has some RSSAC members on it but is not the RSSAC. So I don't feel comfortable speaking for a separate organization at this time. I mean, I could have personal insights from watching from afar, but I think that would be irresponsible to the people who are actually running and serving on the GWG. I know for a fact that it has a number of open meetings this week. Later in the week there are three or four or five sessions. So it's an open session; it would be easy to check in on. But with all due respect, I don't want to speak for the GAC and I don't want to speak for the GWG, so I apologize. I hope that I'm not diverting; it's just, I get asked this a lot and I had been asked by that organization, you don't speak for us, please don't.

EUROPEAN COMMISSION: Let me maybe perhaps try to rephrase. We're just trying to understand a bit better the role of RSSAC. You mentioned in the beginning that it's not a body—not an enforcement or supervisory body, not a representative body for the service operators—and just wanted to understand a bit better, because I know that you've been working on formalizing a bit the governance framework. You mentioned on a few slides in the beginning, just trying to get a better understanding on what you can share about RSSAC, so not about others, of course.

JEFF OSBORN: Okay, I'm going to defer to Rob.

ROBERT GOLDBERG: It's a fascinating process that we've been working through. The role of RSSAC specifically, or the things that RSSAC have done as an institution that have a bearing on this, are the core proposals that RSSAC put forward that describe the needs of a governance structure. The original document that did that, of course, was RSSAC-037, adopted a number of years ago, which described in detail various business processes and various types of decision matrix that needed to be more formalized as the governance structure for the root server system evolves. It was a path and description for evolution.

Similarly, RSSAC has adopted and published a series of core guiding normative principles that form the key framework for governance that was published of 11 principles both in RSSAC 037 and repeated and expanded in RSSAC 055 a few years ago. More recently, in RSSAC 058, the organization published a very long and detailed list of success criteria—what we as an institution felt a governance structure description should include in order to embody success, in order to pull us forward in the right direction. Those recommendations are the extent of RSSAC's—and perhaps some others that one can find in RSSAC published documents—that is the extent of RSSAC's formal role in the current governance discussions.

The cross-community governance working group which has been chartered by the Board includes representatives from each of the 12 root server operators as well as other key stakeholders in the internet community, the ICANN community, liaisons from a number of different sources. So that is a different room where people are discussing the evolution, the next step. And we, speaking personally, feel very positive

about the direction that's taking. And again, like Jeff, I would defer to the leadership of GWG for a more detailed report. I think this is an exciting week for the GWG though, in a positive way. So I hope that that gives you a little more of a flavor.

NICO CABALLERO: Thank you, Rob. Thank you, European Commission, for the question. I have one last question and I'm closing the queue here because we're already running out of time. Go ahead, please.

KHALED ALTARHUNI: Khaled speaking from GAC Libya. Thank you, Jeff, for the informative presentation. I am wondering how does the DNS root server system work when we have two IP addresses—IPv4 and IPv6—for the same domain? So is it using the same database or...? Thank you.

JEFF OSBORN: The short answer is it works very well. It's what's referred to as dual stack in that there are, in the DNS, there are records for IPv4 addresses and IPv6 addresses. For instance, if you literally looked at the—if you read the zone file—maybe I'm the only one in the back of the room who reads zone files—but there is, for instance, an A record is a record for an IPv4 address, a quad A, which is AAAA, is the record for an IPv6. And since the introduction of IPv6, systems around the world are pretty good at getting both of those. All of the root server instances on earth that I can think of present both IPv4 and IPv6 for the asker. So that's

fully implemented by us. And my colleague from Sweden is going to improve on that because he's much smarter than I am.

LARS-JOHAN LIMAN: Thank you. Lars-Johan Liman from Netnod again. No, I'm not smarter than you, but I would like to add that there are two aspects to this. The one you described is the content in the database, and its content—it holds both v4 and v6 information. But the second point is that all root server identities are accessible over IPv6 transport, so that if your computer only uses IPv6, you can still reach the root servers. Not every one of the 1,800, but I'm sure that more than half of them are reachable, and that is well, well, well enough for the moment. Thank you.

KHALED ALTARHUNI: Thank you.

NICO CABALLERO: Thank you very much, Libya, for the question. I hope that addresses your concern. And again, we're absolutely running out of time. Thank you so much, Jeff, Rob, and your fantastic team. A big round of applause for our friends.

**[END OF TRANSCRIPTION]**