ICANN79 | CF – How it Works Session #2: Root Server Operations
Sunday, March 3, 2024 – 1:15 to 2:30 SJU

NICOLAS ANTONIELLO: Hello and welcome to the session about the root server system. My name is Nicolas Antoniello and I will be the participation manager for this session. Please note that this session is being recorded and is governed by the ICANN expected standards of behavior.

During this session, questions and comments will only be read out loud if submitted within the Q&A pod. If you would like to speak during this session, please raise your hand in Zoom. When called upon, virtual participants will be given permission to unmute in Zoom. On-site participants will use a physical microphone to speak. Interpretation for this session will include English, French, Arabic and Spanish. Please state your name or the record and the language you will speak if speaking a language other than English. Please speak at a reasonable piece.

With that, I will hand the floor over to Ken Renard.

KEN RENARD: Thank you. Good afternoon and welcome. Volume good? All right, so we are members of the Root Server System Advisory Committee, RSSAC, and we're here to talk to you about the technical background of how the root server system works and a little bit about RSSAC. Myself, I'm the Vice-Chair of RSSAC. We have our chair right here, Jeff Osborn, Wes Hardaker from ISI and Liman from Netnod. We actually have six of the

twelve root server operators in this room right now. So we're here. Please ask us any questions. So we hope to give you a good idea of how the root server system works and how RSSAC fits into the broader scheme.

So this is our agenda. We'll talk about some background, the technical details of how the root server system works. We'll look at the root server system as it stands today, a little bit about Anycast, which allows us to do some really very big capacity on the root server, then talk about the RSSAC itself, and then on to some evolution of governance.

So go to the next slide and then quickly to the next. So the fundamental identifier on the Internet is really the IP address. This is what computers use to talk to each other. It's a bit string, or we usually think of it as a numerical address. So if you're connected to the Internet, you will have an IP address. So we see in the bottom right hand corner some examples of IP addresses, IP version four and IP version six. So you probably familiar you've seen these before.

The next slide. So why did we do DNS in the first place? Originally, remembering these IP addresses was difficult. It's very easy to confuse similar numbers. Also they change quite often. We see addresses change as you move your service from one company to another, maybe from different locations. And a more modern problem is that addresses may be shared. So multiple websites could be shared on the same IP address. So DNS really helps us with this issue. It's something that we as users of the system can really grab onto and understand.

So next slide. So the primary use of the DNS is this mapping of a name like www-dot-example-dot-org to an IP address. You see an example

there. And when you look at the namespace, this hierarchy of names, you see the root at the top (that's us), and then the top-level domains like dot-uk, dot-org, dot-edu, et cetera. And there's multiple … We call them delegations below that as well.

There's lots of other uses of DNS—for example for finding mail servers, for sending email. IPv6 addresses are also in there, the inverse of addresses. A lot of security records are kept in there, but most of our use is actually the forward mapping of that name to an IPv4 or IPv6 address. So you see that the great statement at the bottom there: a globally distributed, loosely coherent, scalable and dynamic database. So it's one of the largest databases out there that it's distributed and it works quite well and resiliently.

The next slide. So we'll walk through the domain name resolution process. This is really important in understanding where the root server system fits in. So I'm going to take you through this diagram, which is the logical layout of how the DNS works. This is the diagram that you'll see in every textbook, in every presentation. But then we'll tell you a dirty little secret after that about how it actually works. So we'll start here.

The scenario starts when you've got your device, whether it's your mobile device here in the picture, or your server, your laptop, your refrigerator, or whatever that needs to look up a name. And this happens a lot. So it is going to contact what we call and recursive resolver. This is an intermediate machine. This is not the root server. This is not the TLD server. This is an intermediate machine. It's usually provided by your ISP or your organization. It could be in your basement

if you wanted. It could be a big open resolution service like Google DNS or others.

So you ask a question, what is the address of www.example.com? That recursive resolver is going to take that question and find an answer for you. Again, going through a logical step, the first logical piece is that it needs to know the location of the.com server. It asks that question to the root. The next step is it asks that .com server, what example.com? And then so forth. It goes down the chain.

Once that recursive resolver in the middle there gets the answers from each of these pieces, it remembers them in the cache. It's got a memory of all the questions that it's asked. Of course there's limitations on how many entries will fit in that cache. So once it has those answers, it returns the result back to the end device, the mobile device, your laptop. You're done. Now you can contact that system directly.

So now for the dirty little secret. This diagram does not really show you the relative frequency of which you ask these questions to the root.com and example.com. So when you think about this caching, the answer that it got from the root server at the top there, can stick around for up to 48 hours. So if you think about this recursive resolver, all the users of this recursive resolver, all the questions (it could be thousands, millions, billions) that it gets, it only has to ask for.com once every 48 hours because it already knows.

So the magic, the workhorse, of the DNS infrastructure are these recursive resolvers. They're the ones that do all the work, do all, have all the memory, and talk to the users. These authoritative systems on

the right are very important. They need to be there, but they're actually not used as much as many people think.

So the next slide. So we talk more here about caching. Every DNS record has a TTL or Time To Live. This is what the authoritative server says. “This is how long you should keep this in your memory. This is the limitation (how long you can keep it in your memory) before you should ask me again, because maybe it's changed.” I mentioned before the root server system for the TLD records. That's usually 48 hours.

And it's important to know here that the root servers only contain information about how to find top-level domains. That's all that they know. They don't know anything about what's inside of example.com. They don't even know what's inside .com. They just point you to that top-level domain to get the further answers.

Next slide. So some modern refinements of the DNS. The DNS started out in the mid-1980s and there's been a lot of evolution of the Internet since then, and DNS has evolved as well. One of the big things is DNSSEC, the DNS security extensions. These are cryptographic signatures over the DNS data that allow you to authenticate or verify that the information has not been changed from the original source. The original source of this data is not actually the root server system. It's actually the IANA who creates the root zone.

That resolver, that middle system, can use cryptographic methods to verify the integrity of that data that it gets back to make sure that the address, the answer that it got, is in fact correct. So you do not have to trust where you get the data from. You can actually validate those signatures to go all the way back to the original content creator.

Other enhancements are for privacy. So there are several technologies that can enhance privacy. The original specification for DNS did not have any thought about privacy of these records, so that's become a very important issue. A couple of the topics there: query name minimization, the DNS-over-TLS or DNS-over- HTTPS. Those provide encryption of the question and reply from each of those sections, from the end user system, your phone, to the recursive, and in some cases from the recursive to the authoritative.

Another big refinement (and we'll talk about this further) is START Anycast. This allows us to scale the deployment of those authoritative servers greatly in a much easier fashion.

And with that, I'm going to turn it over to Liman to talk about the next set of slides, the root server system today.

LARS-JOHAN LIMAN: Thank you. So I'm Lars-John Liman. I work with Netnod in Stockholm, Sweden, and we operate one of the server identities.

So next slide, please. I'm going to use some specific terms when I talk here, so I would like to go through them first and try to define what they mean so that it'll be easier for you to follow. The root zone is a piece of the DNS database content. I view the DNS as a gigantic database that's distributed over the entire Internet. And the root zone is one piece of the puzzle. And the root zone is the piece that contains the information of which top-level domains exist[,] both ccTLDs and gTLDs alike[.] When you're in these systems, there's no difference between them. The difference is on the policies and registration procedures, not when it

hits the actual technical system. So the root zone is the list of top-level domains and the associated information to be able to reach down to the next layer in the resolution process.

The root server system is the entire set of all the root servers operated by all the root server operators and all the existing pieces that we need to keep this up and running. Root server operator is an entity that operates one or possibly several of these identities, and there are twelve of us. Netnod is one. Army Research Lab is another, and several are present here in the room here and now.

Root server Anycast instance. There are 13 identities, 13 IP addresses of each kind (IPv4, IPv6) that you can send your question to your DNS query to have a response from the root zone. But using Anycast, we as Netnod can operate many copies of using the same IP address. We have roughly in our case, somewhere between 80 and 90 machines spread all over the globe that use the same IP address. And depending on where on the network you happen to be located as a client or as a resolver, you will reach one of them, and the network will make sure that you reaches the nearest one. You reach the nearest one.

The Root Server System Advisory Committee is a committee here within ICANN. It's actually one of the original ones from 1998, and its role is to advise the ICANN Board and the ICANN community, and even the Internet community in general, on matters relating to the root server system. And there will be more talk about that later.

Next slide, please. So the root zone versus the root service system. The root zone is the starting point for lookups where you don't have anything in your cache. The root zone is located in the root servers. So

we load this root zone into the root servers so that the root server is the way to access this data over the Internet. The root zone is managed by … It says ICANN here. I would like to modify that slightly and say the IANA. The IANA is an affiliate of ICANN, so it's part of the ICANN structure. But IANA is the administrative function that actually does the day to day work, where ICANN is the policy system that sets the policies that guides IANA in how to do its work. The root zone is compiled and distributed by the root zone maintainer.

So the IANA will forward changes to the root zone maintainer, and they will make the changes to the zone and sign it with signatures and then make it available to all the root server operators and actually the general public (this is no secret, this zone file), so that we can download it and put it in our servers in a secure way. And then the root servers provide this information to the general Internet over the DNS protocol.

The root service system responds with data from this root zone—and currently, as I said, 13 IPv4 addresses, 13 IPv6 addresses from over 1,500, and you can raise this number. This is a slightly old presentation, so I just looked before we started. The current number is 1,733. Different server machines on the Internet provide this service.

This is a purely technical infrastructure role. The root zone is what we receive from the root zone maintainer. The data that's served from all the root server operators is absolutely identical on all the machines. And there is actually no way that a root server operator can fiddle with the content of the root zone without it being noticed because of the DNSSEC signatures. The root server operators do not have access to the signing key that's used to sign the zone files. So therefore, if we make

any changes to anything, it will be immediately noticed by all validating resolvers on the network, and all kinds of things will happen. So that's not an option. So this is just an infrastructure to carry the information from IANA, the root zone maintainer, to the general public. And the root server system is operated by the root server operators, yes.

Next slide, please. So this is a picture that tries to illustrate how the data flows through the system. At the far left you have a change request coming in from a top-level domain operator, and that can be a ccTLD or a gTLD. They both interact with the IANA. They have slightly different ways into the IANA, I believe, but it is very similar processes. The IANA function will then, after having approved the change, send the update to the root zone maintainer, who will generate a new zone file containing the changed data and make it available through the root server operators, twelve of us. And then each root server operator has its own set of actual servers—80 or 90 in my case. Others have more, some others have fewer. It varies by operator, but in my case it's Netnod's role to make sure that the zone file is copied out to all these servers. We pick it up from the root zone maintainer, and then we make sure that our servers work well with updated data.

And once the data reaches all the server instances on the network, it can be obtained by the resolvers. And that's an automatic process that's totally seamless. So no one will ever see that these things happen. Suddenly there's updated data and things just work.

Next slide please. This is a bit of a history of the root service system. It's actually been around for 40 years by now. It's been operated by volunteer efforts, and you can see that it has grown slightly over the

years, and the last growth happened in 1998. So the current system, as it's operating now, in principle has been around for 25 years. We haven't had much turnover in the set of operators and we try to keep it extremely stable. Stability and resilience are paramount for the roots of operators. We'll be much happier having something that uses slightly older technology as long as it's rock solid and just works. No experiments with our machines. And from 2001 we started to increase the number of instances substantially. And that's when the numbers really took to the skies.

So the changes that we do are we do respond to requests for change in technology. Adding DNSSEC was something that we had to look at carefully. Anycast, of course, was a big thing. IP version six was a new addition that we had to address, and we did so after careful deliberation, testing, doing experiments in separate environments, collaborating, cooperating and finally deploying in our systems. So there is a change process. We just want to make very, very sure that it is all very stable.

Next slide please. This is just a list of the current root server operators and the IP addresses and host names of the servers that the services that they provide. And you can see that there is a great variety of types of organizations here. We have everything from commercial companies to universities to not-for-profits to government agencies. And this is a very big strength of the system, this diversity, because it means that if some organization or even entire type of organization--say that all the commercial companies—would go bankrupt, well, we'd still have a government agencies who operate or not-for-profits. So it makes the system very resilient.

Next slide please. This is just a snapshot from the web page where the root server operators collaborate to give information about the system as a whole. We operate individually, but we do collaborate and cooperate a lot and we cooperate around this web page. You have the address at the bottom and it starts up this map where you can zoom in and click and see where there are root server instances deployed to see if there's anyone near you or the density of deployment in your neighborhood.

Next slide please. We also collaborate a lot around how to operate the service and we've come up with eleven principles that we operate according to. These are part of the seed or the starting point for the Root Service System Governance Working Group, which I think will be mentioned later. And I'm not going to read them all, but I would like to focus on a few.

So next slide please. Principles two and four. Number two is actually, I believe, the most important of them all. We use the same data. We all agree that the data provided by the IANA through the root sign maintainer is the data we serve. No one in the group of root server operator disagrees about this fact, and that's a very important thing. We serve the same thing. Also, the diversity of the root server operations is a strength of the overall system. So we try to put diversity in all possible places, everything from hardware to operating system to software that uses DNS to operating procedures to type of organization to legislation where we are incorporated. We try to keep it diverse so that we don't have any single points of failure.

Next slide, please. Also, a very important principle is that we need to collaborate and engage with our stakeholder communities. We cannot just sit in our black chamber and do this without having any relation to reality, whatever reality is on the Internet. But we need to engage with the various groups that use our service, because if we don't, we don't really know what the expectations are and we cannot really respond to those expectations. So you will see us participate in ICANN meetings, IETF conferences for standardization, various types of local or regional Internet registry meetings, the RIPE meetings or ARIN meetings NANOGs. There are DNS-focused meetings. You will find the roots of our operators there and we are happy to engage. If you have questions, if you want to talk to us, please do.

Also number ten here. This is also, again, part of the diversity. The RSOs must be autonomous and independent, because if we aren't independent, we cannot maintain the diversity. It's very important that we can make our own decisions about how to operate the service.

Next slide, please. There are a couple of myths around the root service system that we try to debunk. One of them is that the root servers control where the Internet traffic goes. That's not at all true. We can tell systems where the destination endpoint is, but we don't control how that system contacts the remote party. So we're like the GPS to tell you the address is over there on the map, but you have to drive there yourself.

Most DNS queries are handled by root server. Definitely not. So you heard Ken describe that caching is vitally important and that's how

most queries are handled. So root servers are only contacted when necessary.

Administration of the root zone and service provision are the same thing. Definitely not. Someone else writes the book. We only ship it to the reader. We don't look at the content. Someone orders a book. Here's your book.

The root server identifiers have special meaning. No ABCD. You saw them in the list. They are all identical. None of them has any kind of precedence over the other, or is more important, or quicker or anything. We actually serve exactly the same data.

And the root server operators conduct operations independently. No, that's not true either. We actually collaborate and cooperate a lot. We meet several times per year and have technical coordination to make sure that the systems run, so to speak, in parallel. But in doing so we maintain our own independence so that we can make our own decisions in order to align the service in a diverse way.

Root server instances receive the entire DNS query. That's not always true. Sometimes they do or sometimes they don't. It depends on actually the one who sends the query. There are modern operational ways where the resolver, who is typically the client to a root server operator, will only send the part of the query that is necessary to know for the root server to be able to respond with the correct response. If your resolver uses that or not, I cannot tell. But there are modern ways to minimize the queries that go to the root servers and then we don't see the rest of the DNS name that's looked up.

And next slide please. If you happen to operate a resolver, there are some things that you can think about to maintain good service to your clients or end users. Make sure that you have root servers within good reach. There are instances on a good high number of places on this globe, and by testing with network utilities you should be able to see how far, and that's measured by milliseconds how far away you can find your root service. And typically there should be three or four nearby where you are.

Do use DNSSEC. Make sure that your resolver is validating the responses and actually uses these signatures that we provide, because that is the way to make sure that you receive the correct data.

Participate in DNS, technical communities. Changes happen to the system and it's always good to talk to others to understand where things are going, which new things could be useful to use in your operations and so on.

And also if you are interested in hosting one of these, Anycast instances in your network or at your Internet exchange points, do come and talk to us. We constantly deploy new instances at various sites and we are very happy to talk to you about that. We all have different plans for how to deploy, so do talk to several of us.

Next slide please. This is just a graph to illustrate that the number of queries that we receive is actually very high. We've had various types of peaks in these curves, and some of them has made the root server operators collectively sit down and say, “Why are we receiving these large volumes of queries? We didn't have that five years ago.” And in

some cases, we actually found a solution and managed to mitigate the problem. That's interesting detective work, I can tell you.

With that, I'm going to hand it over to Wes, I suppose, to explain how Anycast works.

WES HARDAKER: All right, so Anycast is a technology that was actually developed to allow a server close to you to do the responding, or a server close to your resolver to do the responding.

So we go on to the next slide. There's a difference. So unicast used to be the way to do it. And what unicast means is when I send a query out to the Internet, there's only one box out there that knows that address to respond to it and then answers me back. So all the packets go to the same destination. With Anycast, that's not true. With Anycast, when you send a request out to the Internet, the closest box that you are near actually responds. So it's no longer a single source. It's actually many sources that could possibly do that. I will note that near is subjective. It's logically near in the network, not necessarily physically near, but they're typically very closely related.

One of the big advantages is that DDoS attacks in a unicast system go to the single source, and so it can take out that single source, whereas DDoS attacks in an Anycast system only hit the source that are near the attackers. So at worst case, you're taking out a very small fraction of a much, much larger system, so you end up with a regional outage as opposed to a global outage.

Next. So here's a diagram of the unicast example. You can see the green source on the left hand side. There's only sort of one quickest route. It takes three hops to get to that destination that's on the blue.

Next slide. The advantage of Anycast is now that there is a number of locations, and you may be getting to a closer one. Note that the source is actually now closer to a nearby destination. The other one's still there. It's just not serving traffic from that particular source. It's serving traffic from other nodes that are closer to it.

Next. So as I mentioned a second ago, under a DDoS attack, then if an attacker is up over, say, in the upper left, it may be trying to get sending all of the traffic that it possibly can to the root server system, but only one of those little nodes is actually getting it, whereas the person that was the source on the bottom left is still not even noticing that there's an attack going on in the world because it's been isolated. This is used not just in the root server system. Anycast is a technology that's now used extensively in many systems, including many other global DNS systems and some web services and things like that as well. It's worked very, very well.

Next slide. So we're going to shift now from the more technical aspect of how the root server system works. We will take questions at the end, so please do feel free to queue up any ideas or questions you have based on the technical stuff. We're going to go quickly through the ICANN-related advisory committee that is here today. And that's the RSSAC and the RSSAC Caucus.

Next. So what is the RSSAC? The RSSAC is the Root Server System Advisory Committee, and its whole purpose is to advise the ICANN

community and Board members on matters related to the operation, administration, security and integrity of the Internet's root server system. That's a very narrow scope. We really only concentrate on root-server-related concepts. We don't dive into many of the other aspects that the rest of the ICANN community does. It's very narrow.

Next. So what do we do? We are a committee that produces advice. As I've said, if you compare the number of advice we've submitted to the number of advice that many of the other advisory committees have submitted, it's a lot less because, again, our remit is quite a bit lower. We provide that primarily to the Board, but we do provide documentation to other ICANN bodies and organizations involved in sort of the overall DNS business, and we document, as I think Ken said earlier, some of our principles and things like that.

The root server operators are represented inside of RSSAC, but RSSAC itself does not deal with technical operation. Aside from giving tutorials on how the technical operation works, RSSAC is only sort of the advisory committee, and there's a separate group that deals with how we communicate over the individual technical details of how we do things.

Next. The current RSSAC chair is, to my left, Jeff Osborn. And beyond that is Ken Renard from two different organizations. These are our current RSSAC leadership. I am actually the RSSAC liaison to the ICANN Board, so I sit on the ICANN Board as well.

Next slide. The RSSAC consists of primary representatives from each root server operator, so I'm the representative for the University of Southern California. We also have alternate representatives. In case I

can't vote, my alternate can in my backup, and then we have Internet incoming liaisons from other bodies. I'll come back to that in a minute.

The vast majority of the advisory that we write today is actually done through the RSSAC Caucus. We don't want just a small collection of people coming up with important pieces of advice. So almost everything we do actually goes through the RSSAC Caucus, and that's a body of volunteers that are subject matter experts. So they really are people with DNS expertise primarily, or some other form of expertise that can contribute toward our workload. And these members are confirmed by the RSSAC, and they have to post statements of interest of why they want to be in the caucus.

Next. So the liaisons. We have a number of incoming liaisons. So we have a liaison to RSSAC from PTI, the IANA functions operator. We have one from the root zone maintainer, Verisign, who's actually over here. We have one from the Internet Architecture Board, which I actually sit on, but I'm not the representative from the Internet Architecture Board. We have Daniel Migault who sits in that seat currently. And then the Security, Stability and Advisory Committee, or SSAC, as frequently referred to around here, also has a liaison to the RSSAC.

We also have outgoing liaisons, people that reach out to the other organizations. As I mentioned, I'm the liaison to the ICANN Board. We have a liaison to the ICANN NomCom. We have a liaison to the Customer Standing Committee and the Root Zone Evolution and Review committee.

Next. But as I mentioned, the RSSAC Caucus actually does the predominance of our work. We get into many technical discussions

about how to improve the root server system in various ways or other things to discuss. It consists of currently over 100 DNS experts. They all write public statements of interest, and we give credit. So anybody that helped craft a document and work on a particular aspect, the authorship of the RSSAC documents is not the RSSAC. It is actually the caucus members that help produce that. Some are very technical and some are more higher level in scope. So we get different people helping with different documents.

The purpose of the caucus is to get all these DNS experts together and write these publications. And the purpose of putting the list of names and stuff of who contributed is not only to give them credit, but also transparency for who was actually doing the work. And the whole point of the caucus is really a framework for getting the work done. In the days, in the long distance past, the RSSAC did everything, and then we really wanted to increase the scope to allow everybody to participate who had an interest in the root server system.

Next. One of the things that is beyond just the evolution that I talked about about creating the caucus and actually having members come and do the help is there's also the root server system governance evolution.

Next. So right now, there is no process for adding or removing RSOs. There's no process for making a change. And in part, that's a historical artifact of that the person that was doing the reassignments at the time suddenly passed away and left no follow-on plan for what happened in that particular event.

In June of 2018, the RSSAC published RSSAC037, which is a proposed governance model of how to make changes to the root server system as a whole. And included in this is some of the principles that we talked about earlier that we've later documented, we wanted to make sure that what we produced was well founded in why the root server system works the way it was, why it should continue to have some of these foundational principles, like that IANA is the only source of the root data. These types of things are written into that architecture document. The ICANN Board later approved the creation of the Root Server System Governance Working Group. That's the RSS GWG. It's trying to develop a governance model, preserving those same eleven guiding principles, including more diversity into people thinking about it, more collaboration and independence. I sit on the GWG as well as many people, and it's a lot of work. We're spending a lot of time trying to come up with the best governance model for this system.

Eventually, the resulting model will be open for public comment, so everybody can participate and communicate through the typical ICANN public comment process. And the GWG sessions are open to observers at ICANN meetings. We have four this week? Six? Five this week, many hours, and we hold workshops and other things, too, and talk on a regular basis. It's actually a lot of work.

Next. So right now, one of the things that we did in RSSAC037, which is the start of the whole governance model, is we really wanted to identify who are the stakeholders of the RSS, who actually needs the RSS to exist. And besides the obvious answer of the entire Internet, we identified a number of principal people and bodies, and one is, of course, the ICANN community. The ICANN community very much is

dependent on the root server system. But that's not all. The IETF and the IAB, the Internet Engineering Task Force, is responsible for the creation and management of the domain name system, the DNS protocol. So they deal with the technical bits on the wire and how it works. And then the ICANN community helps construct the whole naming system, along with the TLDs and the registration systems that go along with it. And then, of course, the root server operators obviously have a stakeholder interest in the root server system as well.

Next. So the composition of the governance working group that is trying to create sort of the next framework for the governance of the root server system is composed of members of a bunch of different bodies. And in particular, as I mentioned before, our goal was to increase the diversity of all of the opinions. So we have people from the root zone maintainer, we have people from the IETF and the Internet architecture Board, the IAB. We have members from each root server operator. We have members from the registry stakeholders group, we have two liaisons from the Board, we have a liaison from IANA, and we have … I think I just listed them all. And the root zone maintainer. We really do want to make sure we get all the perspectives so we don't design a governance system that does not take into account everybody's needs and responsibilities.

Next. So that's actually it for our presentation today, and we would love to take questions a little bit more just so that when you can walk away from this, there's actually a lot of documentation online. The ICANN web page about the RSSAC has a frequently-asked-questions list. We do get a lot of similar questions, and it's actually just a great document to read top to bottom because it really fills you in on a lot of details about

the root service system that most people don't think about until you need to know it. And we actually have a mailing list called Ask- RSSAC that you can send questions to as well. That was actually referenced on a previous slide.

And then more information on the RSSAC Caucus is, of course, also available on our RSSAC Caucus group list and how to apply. I think you need to, as I mentioned, create a statement of interest and send that off. And the RSSAC membership mailing list is where you send that to, and they're the ones that will pass it on to RSSAC. And as I said, the whole purpose of the caucus is to collect people with a strong background that are able to help participate. So it's not a learning place. It's a place to actually roll up your sleeves and get down to work and do some real work.

NICOLAS ANTONIELLO: If you can speak a little slower.

WES HARDAKER: A little slower. I apologize. Unfortunately, I'm done. I'd be happy to do it over slower if we have time.

Does anybody here have questions about anything that we have discussed, about anything about the root server system or something that you think that we have missed? We would be happy—

NICOLAS ANTONIELLO: If anyone in the room or remotely in the zoom session would like to make any question the Q&A pod, it's open for that. Or you can raise your hand and we'll allow you to turn on the mic to do the question.

WES HARDAKER: No question is too small.

NICOLAS ANTONIELLO: I have some questions that I've collected through our engagement instances that are frequently asked, and as we have the most expert and direct operators here, maybe it would be good to listen from you for those.

One of those questions that we frequently have is that you mentioned that each time a resolver learns something new from an authoritative server, beyond giving back the response to the device that made the question, it will store it in the memory and will put it in the cache for a while, right? And then after that time it will eventually get that information. Who gets to decide how long that information may live in the cache?

WES HARDAKER: That's an excellent question, and it's often misunderstood for a couple of reasons. One, every record has what's called a time to live. How long are you allowed to memorize it? That doesn't mean you have to, right? It's the maximum of time that you're allowed to. If you don't have much memory on your device, you may forget it faster. And the IETF has actually produced a document lately that says if you can't contact the

server, you can actually use it longer. So even that is no longer a maximum, like it was always said. But every different piece of the DNS system sets their own values. The root zone is the longest that I know of. Well, that's not true. I've seen longer ones, but two days is the limit, and so software will remember answers from the root zone for two days at times. Other websites like to change their addresses on a very frequent basis because they're doing load balancing and stuff, so they may have it set to five minutes or even shorter.

As a piece of advice from I work for a university and my colleagues at the university have studied this, and longer timeouts mean that you are protected against attacks better. So you want to find the balance in there. And often what we recommend is an hour or a day, even, depending on your needs. But the root server system? Two days.

NICOLAS ANTONIELLO: And as a kind of follow-up to that answer, are you aware if there's any ongoing work to allow a registrant of a domain, for example, or the one that owns the name to modify the TTL?

WES HARDAKER: That is a great follow-on question, and it's unfortunately not the easiest answer because there's actually two different places to get some of that information. When you register with a TLD or another registry, they get to set some of the time-to-lives, but you get to set everything in your domain. So your www record or your mail record or things like that are under your control. But the nameserver record from your parent is often set by the parent, not by you.

NICOLAS ANTONIELLO: And I have another one which is completely different, which is frequently asked. Also, when you explain the way that the root zone file is transmitted from IANA, if I didn't get it wrong from you, [who is] the only one that can modify and the one that produced the file, then it's transmitted to the root zone maintainer and then to the root zone operators who are the one that runs the server that make it available to the general Internet, let's say.

So suppose the very unlikely event that root server operators lose contact, technically speaking, with the roots zone maintainer, so they cannot get the updates from the roots zone maintainer. How long in practice will the root server system continue to work, to operate? How long the root server operators will continue to serve the root zone until something is said, like. “Okay, we cannot go on with this,” if any? Thanks.

LARS-JOHAN LIMAN: I can take a stab at that. A server at the end of the line, one of these instances, will continue to operate with the same data until it realizes that the change has happened to the zone file. So it needs a new copy. And that can happen in two ways. Either it receives a poke (it's called a notification message) that says, “Hey, there's a new version. Come and pick it up.” Or it has a timer internally that says “Every (whatever the number is) 2 hours, I should go and talk to my mothership and check if there is a new copy of the file.” And the same mechanism is used in all the various steps when this zone file is transported. So if you have a root

server operator, they have a couple of central distribution facilities for their own armada of instances.

And the problem you describe is if that mothership cannot talk to the roots or maintainer anymore. If we step back a little, that means that we still have a functioning relationship between the instances and the mothership. So there we have an ongoing ticking saying, “Has anything new happened? Has anything new happened?” And the mothership is not aware of anything new happened. So it will respond to you, “Everything is fine. Everything is fine.” And the end instances at the end of the line will continue to operate as normal until another timer kicks in and says that, “No, now we know that the information is too old. And now I will have to look this up on my computer here, actually, to get the actual number.” And that one was off the line, so—

WES HARDAKER: While you're looking that up, Liman, can I suggest one other thing, which is that one thing we don't explain well in the slides is the architecture of the root zone maintainer. They use Anycast. They use a repeated system. I don't know. Duane is our root zone maintainer volunteer. Do you want to talk of anything about your infrastructure? We'll just throw it over to you.

DUANE WESSELS: Sure. Thanks, Wes. So, yeah, I'm Duane Wessels. I'm the root zone maintainer liaison to RSSAC. So, yeah, when Verisign publishes the root zone, it goes out to hundreds of Anycast instances. So the probability that that communication would be lost or broken is very low. But if you

want to imagine what would happen if that did occur, then I think the answer to your question is one week. There's a value in the SOA record called the expire field, which for the root zone is set to one week. And DNSSEC signatures also come into play, but we make sure that the signatures are longer than the expire time.

LARS-JOHAN LIMAN: I would argue that the numbers in there are correct, as you report them. But you actually have this two-step procedure which doubles that time. So you will have two weeks. Our mothership will time out after one week, and then we have another week until the instance times out. And that matches with the signatures, as you say. So I would argue two weeks.

DUANE WESSELS: Eh, I don't know.

[WES HARDAKER]: Let's not get into detail.

LARS-JOHAN LIMAN: Okay. So we have one to two weeks to fix the problem. And as soon as it appears, you will have dozens of engineers throwing all their working time at fixing this problem. And if it really comes down to not being able to fix it, there are actually other ways to transport the zone file. We don't have to use this Internet to transport zone file. We can have someone load it onto a thumb drive, a USB drive, jump onto an airplane and fly

to Sweden and drop it in my lap. And there's the new zone file. And we can do that. It's been much quicker than a week.

DUANE WESSELS: Carrier pigeons.

LARS-JOHAN LIMAN: Yeah, carrier pigeons. You can do it by submarine. You can do … Just imagine. We don't have to use the Internet to transport this. And as soon as we notice (and we notice that quickly) that these things don't work, we have monitoring system with all the root server operators, and there is a ton of people out there monitoring our systems from the outside. So we will know when it happens, if it happens. But even so, we have at least a week to fix that problem.

WES HARDAKER: The important thing is that it has never happened since the creation of the root system. Not once.

[JEFF OSBORN]: I have to relate something. I'm fortunate enough to work at a company where my Board of directors, between them, have something crazy like 200 years of experience. The Internet is three members of the Internet Hall of Fame. And I brought to them the question. I said, envision the failure of the root server system. And then one of them sat back and went, :Oh, I've got it in the fourth month of nuclear winter …" And I was like, what? It is shockingly difficult to envision the system failing. I mean, two weeks is a long time when everybody's trying to fix

something. To fix it, you could literally put thumb drives on goats and send them in the right direction. It's a shockingly long time to fix things, and then all that's broken is you simply wouldn't have gotten the new addresses for things. Everything existing is still there. It's a remarkably resilient system.

UNIDENTIFIED MALE: And one thing to note is we're all nerdy engineers and we think about these things all the time. We monitor the heck out of this system as well as other people do. So these are all disaster scenarios that we've discussed for a long time—well, not with the nuclear side. But this goes into the engineering. It's been built into the system. These securities, stability and resiliency has been built into the system from the start, and it continues to evolve.

I think there was a question in the back.

UNIDENTIFIED MALE: Hello. Considering the number of ccTLDs and gTLDs over the years has been quite steady—not recently, but yeah, during the initial phases of the Internet—in your experience, do you think that the resolvers also always go to the root servers for the initial dot every time, [for whatever] mistake that is written on a browser, for TLDs that don't exist, for example, or the resolvers simply filter them out themselves?

UNIDENTFIED MALE: I'll take this one. We see a lot of queries for something.pdf. So yes, there's a lot of mistakes. Duane is at about 50%. About 50% of the

queries we get are obviously from broken systems. So that's out there. It's a huge volume and it's still just a small blip on our radar. The capacity of the system … You can have stuff that's been misconfigured. If you look at that usage graph, there was a system that purposely put out essentially garbage into queries and we saw it, we recognized it, and we had fun being our nerdy selves and looking into it, but it did not affect anything about the capacity of the system. Thanks.

WES HARDAKER: One additional point. There's a number of technologies to actually improve upon negative results. So what you're really talking about is when you query for something that doesn't exist in the root, there are two things. One is (I don't want to get into the details of the cryptography) aggressive NSEC caching. Negative caching is something that has been used where many global resolvers will actually know that there is nothing between apple and carrot, so that Bob doesn't exist and it won't even ask. So that is being more and more widely deployed these days as well as there's technologies to actually keep a copy of the root zone yourself.

I actually run a project out of ISI called Local Root, which actually helps you configure your local resolver to have a copy of it itself. You can't serve it to the rest of the world, but you can serve it to your own organization, for example. And there's an IETF RFC called RFC 0886 that talks about that too. That also helps with those negative answers because you can answer them right where you are instead.

NICOLAS ANTONIELLO: We have some online questions. We have some questions made by [Astrid Neges]. Question number one is, can you give us a general description of the requirements of managing a root server? We have ten minutes only.

LARS-JOHAN LIMAN: I actually made a list in another presentation I gave a totally different [inaudible]. To operate a root server, well, you have to have really high-level DNS expertise. As you mentioned, Jeff, we're talking on combined numbers here in hundreds of years operating DNS servers. And you need to have a good standing in the Internet community. People have to know who you are so that you can really prove that you have this expertise and what you need.

You need financial stability. This costs money. We don't receive any money from anyone for providing this operation. This is a pro bono thing that each root server operator provides for free to the community at the point where it's consumed. It's not for free for us. It costs real money. And you have to have an idea of how to sustain that cost.

And you have to have the technical expertise in operating systems, both in knowing how to run servers, because the uptime requirements are not 99.99. It's 100.0.0.0.0. The system must not fail.

And then you also have to have network experts, because this Anycast thing is no easy beast. It puts a lot of interesting properties into how network traffic flows and how you respond to it and where you send responses to and whatnot. It's actually rather complicated. So you have to have your networking expertise in place as well.

And it's always good to know the guys who write the software so you can call them in the middle of the night and say, “Hey, I'm one of your friendly root server operators, and your software has a bug. Can you please fix it?” “Yes, 15 minutes will be fine.”

So being a part of the community is a very important thing and also have a position where you are able to follow these principles, the eleven basic principles, so that you can be neutral, that you can be impartial to all the clients, where these queries stem from. So there is a good number of things that you have fulfilled and I have probably forgotten five.

[JEFF OSBORN]: I'll mention another aspect of that question is what it takes to be a root server. There's ongoing discussions now about how do you become a new root server operator. So one of the big aspects of that is, is there a need for another root server operator? So that's being discussed in the governance working group as we, we talked about. What's the right number? We don't know. Is it 13, is it 100, is it five, is it one? So those aspects down the road—how to add and remove root zipper operators—are being deliberated now in the governance working group and will be open for public comment soon.

NICOLAS ANTONIELLO: Thanks. Following question is: the propagation time used by a … Sorry, again, following … There's one question before that one. So how much do you think is the latency cost by not having a root server nearby to your location?

WES HARDAKER: Very minimal because you're not talking to the root server system frequently. So you will not notice. If you are the unlucky person that makes the first request for your favorite website, and nobody else near you has made it for an hour or more (I mean two days), you're not actually going to go talk to the root again. If you are the first person in two days to make a request for .com, it's going to be 50 milliseconds—100 milliseconds most likely, at worst. You're not going to notice.

[JEFF OSBORN]: If I can just plug in because it was a surprise to me. We found out in investigating the report that I gave that was similar to this earlier today (and I'm giving it again in about an hour) that it's about one out of 5,000 to 10,000 queries that require a trip to the root server--one out of 5,000 or 10,000. So the vast majority of queries are answered out of the cache, and then many of the rest of them go to other authoritative servers long before they have to get to the root server. I knew it was a big number, but I was really surprised. It looks like it's a very credible 5,000 to 10,000. And again, if you're introducing 15 to 100 milliseconds into a search, I certainly wouldn't notice that. I mean, that's like lag you notice in a game. That's hardly something where you hit return and it refreshed 15 milliseconds later than you're used to. It's a shockingly small amount of time.

LARS-JOHAN LIMAN: So the number that's important is the distance between yourself, your laptop, and your resolver, not between the resolver and the root name

server. And there you might want to pay some attention. But again, as you said, we were talking milliseconds either way.

NICOLAS ANTONIELLO: There's another one. I'm going to read it just—okay.

EUGENIO PINTO: Sorry, another question.

NICOLAS ANTONIELLO: Can you state your name just for—

EUGENIO PINTO: I'm Eugenio Pinto from the dot-pt registry. My question is, with the advent of the blockchain, can you envisage a world where the root information would be published on a blockchain instead of the current system?

WES HARDAKER: So I actually run a blockchain-related working group in the IETF. Not about the DNS though, but about something else. And I've written papers on blockchain. So the funny thing about the DNS route, as he said, is it could be over carrier pigeon, it could be over a USB drive. It doesn't matter because it's signed. It's signed data. So no matter where you get it, if I hand you a copy on paper and you go type it in perfectly, it'll actually work. So the distribution method for the data actually doesn't matter. And I suspect there probably is a blockchain out there today with the root data somewhere in it, because we seem to be

putting everything in write-only logs. It's an excellent technology. I've used it for other purposes. Whether that takes over the distribution mechanism is yet to be determined.

As I said, my local root project allows you to take the data at the same time as everybody else gets it and then run a local copy. “Why does it need to be in a blockchain?” would be typically the question I'd ask. What benefit are you getting out of it? Because you're already getting signed, non-mod-viable data. So it's just another transport mechanism at that point. Good question, though. Thank you.

NICOLAS ANTONIELLO: There's another one in the session. It says, is the propagation time used by a registrar to a register a new domain name affected by how far you have a root server?

LARS-JOHAN LIMAN: No, not at all. Not related in any way, because that's a transaction that happens between the authoritative name server further down in the tree. So we're only talking about what's the propagation time of a registration of a top-level domain. There the root server could be involved, but that's a very slow and deliberate process. So I would argue it's good if it's slow, because that means that we can keep track of everything that's happening and that there are no false things or dangerous things happening during that path.

NICOLAS ANTONIELLO: Okay, there are no more questions in the Q&A. There's one in the pod? Yeah, I asked the one that made that question to reformulate, please, because I didn't get to understand correctly the question. So far it's not reformulated. I'm going to read it anyway. But if you can understand, because I cannot: considering the potential of artificial intelligence to efficiently process large volumes of data, would it be feasible to explore a collaboration between AI and root servers, particularly those maintained by organizations such as NASA, to enhance the collection and analysis of meteorological data?

JEFF OSBORN: No.

WES HARDAKER: Again, thank you. Jeff. I always ask the question, why? What is the benefit of doing that unless you have a reason? I mean, AI is a great tool. I love playing with it, but unless you have a goal, it's not an intelligent entity that can come up with a new technology. AI is not at the point of doing invention yet. So what question would you ask it about the root zone that it could answer that would be new?

JEFF OSBORN: That had anything to do with weather. Well, I'm sorry, this is a no.

NICOLAS ANTONIELLO: Seems that there are no more questions in the Q&A pod. If there are no more questions in the room, I would love to say thanks very much to all of you for this amazing presentation. It was very nice to have you all

here and be able to ask you any question. So thanks again. Thanks very much. So the recording can stop now. And thanks very much. Meeting adjourned.

[END OF TRANSCRIPTION]