ICANN84 | AGM – How it Works: Root Server Operations
Tuesday, October 28, 2025 – 16:30 to 17:30 IST

YAZID AKANHO

Good morning, good afternoon, good evening, everyone. Although for those who are in the room, it's, of course, good afternoon, everyone. But we do have remote participants as well. My name is Yazid Akanho. I work for ICANN Technical Engagement. It's always a pleasure to introduce this session which is the How It Works: Root Server Operations at ICANN84 here in Dublin. I'm the participation manager for this session. Please note that this session is being recorded and is governed by the ICANN Community Participant Code of Conduct, ICANN Expected Standards of Behavior, and the ICANN Community Anti-Harassment Policy.

Please observe the following guidelines to participate in this session. I will also post them in the chat for your reference. Only questions posted in the Q&A pod will be read aloud during this session as time permits and when directed by the chair of the session. If you wish to speak, please raise your hand in the Zoom or otherwise as directed. When speaking, please state your name for the record and speak clearly at a moderate pace.

Today, presenters will be Ken Renard from U.S. Army, DEVCOM Army Research Laboratory; Lars-Johan Liman from Netnod; and Rob Carolina from ISC. With that, I'm happy to hand over the floor to—oh, Rob. Okay, good. Thanks. Rob, over to you.

ROBERT CAROLINA

Thank you very much. Just to make sure, am I close enough to the microphone? Can I be heard? Is everybody happy? Is there someplace else you'd rather be today? I don't think so, because this is the place to be today. Thank you very much for this opportunity to talk to you. I'm just making a little adjustment here to see how far I've gone.

Our challenge today is to explain the root server system itself, and in doing that, I must confess that the three of us, of course, are members of the Root Server System Advisory Committee. Each of us works for a root server operator, three different ones in our cases, and out of the three of us, I will immediately let you know something that you will quickly find out anyway, which is I am the least technically competent person in this panel, quite possibly in this room, potentially in the entire building. But that remains to be seen, and that's because I am a lawyer, I'm general counsel with ISC, but I've been spending a lot of time in the last five years getting to grips with policy and regulatory and business operational needs of the root server system, as well as the technical aspects of DNS, so I bring a slightly different approach to it.

I'll be presenting the first dozen or so slides, and then my colleagues will start filling in with even better information. So to move straight into it, if I can—there we go—get the slide deck moving. The agenda for today, the Domain Name System, I'll be talking about that. Then when I'm done talking about that,

generally, the anycast, the root server system today, RSSAC and the RSSAC Caucus, the two ICANN bodies that you will have heard about, various principles and governance, and the root server system and you.

To begin, the Domain Name System. Welcome to ICANN. This is the thing we were going to talk about always and ever because it is a key linchpin of what people do at ICANN and how the Internet works. Keep in mind that the Domain Name System functions at its base is meant to do one thing exceedingly well, and that is to take a domain name, a string of alphanumeric text and numbers, and map it to, translate it to, give information about the IP address associated with that domain name. That's the purpose of the DNS is to do that mapping function.

Those of us who are old enough to remember telephone directories can think of it in the sense of looking at an entry which is a name of a person or a business, and at the other end of the side of the page is a number. It's similar to that kind of mapping, except that it's done on a massive global scale.

IP addresses, you're familiar with, IPv4 or IPv6 192.0.2.1 is a v4 address, and the one underneath, which I won't attempt to say out loud, is a v6 address. Suffice it to say v6 is longer, bigger, more complicated, and that's a good thing. But I'm only going to speak to v4 addresses because they're easy to talk about. The principles are the same.

Now, the original problem of DNS, thing that was designed to resolve a couple of problems, two significant ones. There are a few others maybe. One is that numbers are hard to remember. IP addresses are hard to remember. Although I had a job 40 years ago where we had a terminal, and no one explained to us what it was, but you could use it to remotely open a door or remotely lock a facility. They said, "What do you do in order to unlock that door?" You say, "What you do is you type in 192.168.1.52, and hit Return." "Well, what does that mean?" "You don't need to know. Just type those numbers and the door will unlock. And you type them again, and the door will lock." We had no idea we were on a LAN using private IP addresses. So this goes back a long way in time. IP addresses are hard to remember, and thank God we didn't have to remember more than one of them.

But these days especially, IP addresses change often, by which I mean, if you are using a device or you have an end device like a server, you want to be able to place that server in different places. You might want to put it on your network, you might want to take it to a colocation facility, you might want to take it to a service provider facility without having to change its name as we see with a DNS with a text string. You might be changing its IP address a lot. And what the DNS does, what the system does, again, it maps the name to the number. And you can change numbers, and some people do change numbers frequently. Some numbers don't change very often at all. Address numbers don't change very often

at all. Some change very frequently indeed. Apparently, some change too frequently, but that's another story.

Some more modern problems that I'm not going to touch on certainly is that IP addresses may be shared amongst more than one resource, and multiple IP addresses may serve as entry points to a single service, and which ones do you use? But all of those are folded into—I'm only staying with the simple case for the moment. If you have questions about those others, please raise them in the Q&A, and I'll make one of my friends here answer them.

In describing DNS, we describe the namespace, there is the root level, top level, second level, third level. So, for example, a second-level domain name, as you can see from this chart, would be something like icann.org. A third level might be something like www.icann.org. Each of those can be called a domain name, or in the UK, where I live, bbc.co.uk is technically a third-level domain name. And those levels can continue to a fourth level, a fifth level, a sixth level. There's a tremendous amount of flexibility given to the person who has registered the second-level domain or the third-level domain with the registrar, a tremendous amount of flexibility to create multiple levels and to create multiple identities on a single level.

I'm familiar with a company, for example—I won't name them—that's in the software as a service business, and they have their main second-level domain, bigbigcompany.com, but then for each of their customer installations, they hand out another third-level

domain, ibm.bigbigcompany.com and somebodyelse.bigbigcompany.com. And they have like a half a million of them. That's the kind of flexibility DNS brings, the ability to create all of these things, and it gives that power to the person who's registered the relevant domain.

Now, in this, I'm going to tell you how DNS resolution works, and then I'm going to tell you why it's all a lie. But first, the lesson that you would get if you were sitting through a computer science lecture at my university, where I teach part time, or most of the universities and most computer scientists and almost everybody else, in fact, this slide teaches it like this. It says in the world, you have an end user device, like a phone, like a desktop, like whatever it is, and it has to find something, it has to connect to something online. It says, "I need to connect to www.example.com." How do I do that? How do I complete that?

It asks something called a recursive resolver. Now, in the world of DNS, keep in mind there are recursive resolvers and something called an authoritative server. They both have a critical role to play in DNS. The recursive resolver is the piece of equipment, or piece of software, or piece of functionality which can reside in a lot of different places that answers this question. The end user device says, or the device says, "I would like to connect to www.example.com. What is the IP address I need to connect with?" The recursive resolver says, "Hang on a minute," or more likely, "Hang on three milliseconds, and I'll find that for you."

Now, the standard answer about what does the recursive resolver do? It does all the work. What kind of work does it do? In theory, the first thing it does is it makes a call to the root server system. Hey, now you know you're in the right meeting. We said we'd tell you about the root server system, and there it is on slide nine. It only took us this long. Very first call to the root server system. So here we have a root authoritative server. The recursive resolver says, "Hey, I'm trying to find www.example.com. I don't know where to go. What can you help me with? What do you have on tap?"

The root server then says, "Well, I don't know the answer to your full question, but I know who you can talk to. You should talk to the .com authoritative server." "Here is the IP address for the .com authoritative server," is the answer that's given back. So the recursive resolver is not very nice or kind. It doesn't say thank you very much, even though it should.

It then places another call to the .com authoritative nameserver and says, "Hey, can you help me find www.example.com?" Once again, the .com authoritative nameserver says, "Well, I don't know the exact answer to that, but I do know where the authoritative nameserver for the registered domain, example.com, is located, what their address is, and here it is." "Well, thank you very much."

Fine. The recursive resolver makes a third call and says, "Hey, I'm told that you have the authoritative directory for everything@example.com. Where is www.example.com?" And the authoritative nameserver says, "Glad you called. The answer is

93.184.216.34." And to complete the picture, the recursive resolver then answers and says, "The address you're looking for is 93.184.216.34." So there's another link back to the original piece of equipment with that address detail.

That is theoretical way that all of this works, and I'll tell you in just second why it's a lie most of the time. That's because of caching. This recursive resolver, this thing in the middle, actually has a memory. When it learns addresses, it memorizes them and keeps them for, let's call it, a while. How long? A while. We'll talk about that in a minute. It keeps them for a while. So if it already knows the answer to one of these questions, it doesn't ask the question. So recursive resolver remembers answers to queries they receive. Every DNS record has a time-to-live value. I said it keeps them for a while. How long is that? That's the TTL value. That's the TTL value. I believe that for root zone records, the typical TTL value is 48 hours. Now, 48 hours doesn't sound like a long time, but when you multiply that into milliseconds, it's an eternity.

Root servers, the one thing I need to emphasize, if we look back at this picture again, the information on that root authoritative server, what is the information it's serving up? Something we call the root zone. And the root zone consists of addressing information so that you can find top-level domain authoritative servers. Top-level domain authoritative servers, that line in the middle. How many of those are there in the world? About 1500, approximately. It numbers a little south of that, or a little north of that, something like that. But that's it. Now, for each of those, it actually has about

10 or 20 pieces of information because engineers like to make everything complicated. But nonetheless, it's a relatively small file, and it's a relatively small amount of data that's on offer at the root server system.

Finally, the last thing that I'm going to speak to for now, when you look at modern refinements to DNS, DNSSEC, which I'm not going to comment on that too much, I do want to take a moment to comment on this middle one, privacy, or in London where I live, privacy. I don't know. My accent has gone all over the place. One of the challenges that you find is you look back at this picture and eventually people realize, "Hey, wait a minute. Why does the recursive resolver need to ask the root server system if the only thing it needs is how do I find the .com authoritative nameserver? Why am I asking, 'How do I find www.example.com?' Why should I ask all of that?" And from the perspective of one root server operator, we'd like to know, why are you giving us all this information? We don't want it. We don't need it. We flush our logs very quickly because we want to hold on to it.

So there's a protocol out there called QNAME Minimization which says the recursive resolver can be configured in such a way that it only asks for the one piece of address information it needs. The one piece of address information that it needs. So instead of asking, "How do I find www.example.com," it could just say, "How do I find the authoritative nameserver for .com?"

And in that way, the recursive resolver is not disclosing any more information about the query string to the root authoritative server. And similarly, at .com QNAME Minimization would say, "How do I find example.com?" It wouldn't have to disclose that it was looking at www. It would only disclose that part, QNAME Minimization. It's a great protocol. It limits the amount of disclosure.

Here's the issue. Not very many recursive resolvers in the world implement it. It is, by no means, the default. The simplest thing is what often happens, which is the entire search string goes out. So just something for you to be aware of.

Resilience. Here's my last comment. Sorry, I'm getting over something. Actually, that's all I've got to say. I'm going to move on because I'm just stalling at this point. So, Ken?

KENNETH RENARD Thanks, Rob. My name is Ken Renard, Army Research Lab, one of the root server operators. We'll continue on with this discussion. Rob made a good point about we don't always have to talk to the root server system. Because of that cache in the recursive resolver, we don't actually rely on that root server system all that often. I want to make a statement about how important the root server system is and how important it's not. We like to say that the root server system is important, but not that important.

Information that comes from the root server system is needed in every single transaction, but it's almost always coming from that

cache. So it's important to have that information when you need it, but the root server system itself, you don't actually have to talk to it every single time. That said, the root server system is definitely important because if it were to completely go away, then all this DNS resolution stops, and basically the Internet, as we know it, fails to work. So having that availability of the root server system is very important.

We're going to continue on about technology and talk about anycast. Typically, what we think of when two computers are talking to each other, it's something called unicast, one source and one destination. Packets from one source all go to the same destination, and in the reverse, the same thing. So that a single computer with a single IP address is the end point of your traffic.

Now, if you have a denial-of-service attack, that means somebody is maliciously sending a lot of traffic to your server, essentially, all of that information, all that that traffic, is going to one computer in one location. That can be pretty risky. On the other side, we have anycast. So where we have multiple instances that can serve the same data. What we end up with is multiple computers that all have the same IP address where any one of them can serve the exact same information. So depending on where you are on the Internet, it depends on which of those several anycast instances, you will actually end up talking to. In a denial-of-service instance, like your case, we have more than one computer that can sink the traffic, that can be the destination of those packets, such that a denial-of-

service attack, often becomes localized or is it can be much better handled.

So we'll look at this pictorially. Look at the right clicker. So here's unicast. We have our source on the left, destination there on the bottom right. One source, one destination. And we try to take the shortest path between the source and destination.

In anycast, we have multiple—is that blue or purple? I'm color blind. Multiple locations that you could go to, but that source ends up talking to the closest destination. It could use any one of those, but the routing of the Internet has chosen to route those packets from that source to that one of three destinations. So that gives you potentially shorter path and allows you to have resilience in case of an outage. If that destination on the lower part of that diagram goes away, maybe it's under maintenance or being rebooted for patches, that traffic will go to a different location.

Under denial-of-service attack. We see in this diagram a denial-of-service attacker that's in—we'll call this regional—that traffic is generally going to go to the same sink, essentially the same anycast location, which you can see that source and the destination on the bottom that traffic is not affected. Only one of those destination nodes in this scenario is affected. So anycast will help, not completely eliminate, denial of service but it will significantly help.

Okay. Looking at the root server system today, few definitions here. IANA, the Internet Assigned Numbers Authority is a function that manages the root zone. So I think it's going to be on the next slide.

The root zone is the actual information, the database, the list of top-level domains and their addresses and associated information. The root zone is that top level of the DNS hierarchy. The root server system is just the set of computers that serve that zone via the DNS protocol. So, we, the root server system, we're kind of like the IT department of the zone. We don't control the contents of the zone. We just get a copy of it when it's updated, we put it on our DNS servers, and operate them and answer questions.

A few more definitions here. A root server operator is an organization, a company. There's 12 of them that manage the root servers, and they're responsible for operating these servers. Again, we don't control the contents. We just operate those computers around the world that do this. We each operate multiple root server instances, so we each have our own IP address, v4 and v6, and we have multiple anycast instances around the world. So, one IP address, multiple locations, just like we had in the previous diagram here. And then last is the Root Server System Advisory Committee that's comprised of root server operators as well as some liaisons. We operate within the ICANN ecosystem, and we provide advice to the ICANN Board in the community about the root server system.

Let's see what else is here. Okay. This is the list of root servers. These are our IP addresses, completely public. This is what we do. Your recursive resolver, out of the box, knows all of this information. This is the information that that software already knows and has to enable that first step of that resolution process.

If and when it needs to contact a root server instance or anything in the root server system, this is the basis for finding that.

Let's see what else is in here. Okay. A little bit of history here. DNS is over 40 years old. It's been around for a while. It started at the University of Southern California Information Sciences Institute. There were initially two servers, but by 1985 there were four. With the growth in the mid to late '80s, it spread to a few more academic institutions—before it was called the Internet—grew. And by '91 it spread outside of the United States, to Lehman's group. And by 1998, that was the last time that 13 root servers actually meant 13 physical computers. In 1998, there were 13 physical computers running this root server system. That would not cut it today.

The reason we talked about anycast is really about the scalability, because it explains how we scale the root server system. Up until 1998, we just added root servers, added new addresses. Actually, it's 2000 when we started introducing anycast, we had an entirely new dimension in which to grow the system and scale it. So, now with the same 13 root server identifiers, the same 13 addresses, we have almost 2000 instances around the world. It was 1999 as of this morning. Any day now, it's going to go roll over to 2000. So, that's the important piece, especially about anycast, is that while we have that list of 13, it can grow in an entirely new dimension and scale significantly.

There is our website there, root-servers.org. You can go and you can look at it. You can zoom and pan around this map here and find

locations of each of those 2000 instances around the globe. It's fun to look at some time and there's all kinds of cool measurements as well. Here's one of those cool measurements, looking at the number of queries received. So this is actually the daily average, I guess, aggregated by week. On the right side of this, 2025, we're getting about 125 billion queries per day among all the addresses, all 2000 instances. So, we see this growth. Our infrastructure, those 2000 hosts, we don't scale it for these numbers. We scale it for the worst case scenario, when we're under attack, that's the capacity that we handle. So this level of queries here, 125 billion queries a day, is basically noise. It barely pushes the needle for us.

LARS-JOHAN LIMAN Another day in the office.

KENNETH RENARD Yeah, another day in the office. Another 125 billion queries.

All right, a few myths about the root server system and some of the reality. The root server system is often misunderstood. Here's just a touch on a few of these things. Technically, we do not control where traffic goes. We only give you the location of that top-level domain nameserver. So routers actually control where your data goes. We're just one part of the address lookup.

Let's see. Again, the administration of the root zone is actually not handled by the root server operators. We don't control the

contents of the zone. We just get a copy of that database and serve it on our computers.

There are only 13 root servers is a myth. It was true back in the late '90s. No longer. There's almost 2000. We as root server operators, we are independent organizations. We have independent infrastructures, and that's on purpose. We have diversity in our implementations. We use different hardware, we use different software, we use different management techniques for our systems, and that's a good thing. Because a failure in any of those things, if we all ran the same brand of computers and there was a problem with that brand of computer, that could be disastrous for the root server system.

So we work together. We work hard to measure and promote that technical diversity. We also have diversity of organizations. Some of us are government organizations, some are commercial entities, nonprofits, universities. So a failure technically, or a failure organizationally, or a failure of an economy or market space is not going to take down the root server system. And that's really by design.

Okay. Again, trying to do differentiate the root zone, the information, the database, from the root server system. Again, the IT department where the computers that make it available via the DNS protocol.

Let's see. One more kind of cool thing here. When you think about the entire root zone and how things are managed, say, for example,

your TLD operator .com is in the room. We have several TLDs walking the halls here. If they need to change something, they go to IANA. They make the change. Maybe they change their nameservers. When the next round of generic top-level domains happens, that will be an IANA Function responsibility as well. That change is made in the database. The root zone maintainer is responsible for wrapping this up, doing the cryptographic operations, and they publish what they call the root zone. It's basically a text file about one megabyte. It's pretty small. It's 1500 or so entries.

That's when it gets to the root server system. It goes out to each of the operators. Each of the operators send it out to all of their 2000 instances total, and that's where those DNS resolvers come in. So in this whole process of whether you're updating a TLD or the next round when we add TLDs, the root server system is this kind of narrow spot right in there. With that, I think we'll hand it off to Liman.

LARS-JOHAN LIMAN Thank you. I'm Lars-Johan Liman from Netnod, another root server operator. I'm going to talk a little about how the root server system and the root server operators, in particular, function inside the ICANN ecosystem. So there are a couple of organizations or sub-organizations inside ICANN that will be beyond on display here. The first one is RSSAC, the Root Server System Advisory Committee and the RSSAC Caucus, which is a sub part of that. So let's dive in.

The role of the Root Server System Advisory Committee 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. That's a text copied out of ICANN's Bylaws. That is the role that RSSAC has to fill. This is a very narrow scope.

What RSSAC does, RSSAC is a committee, and we produce advice—that's the role we have—primarily to the Board, but also to the other ICANN bodies. For instance, if policy-related stuff inside ICANN risks having an impact on the root server system. For instance, when the first round of new gTLDs was being proposed back in 2010 or so, we undertook a rather big study. How would that affect the root server system if we added 1000 or 2000 new top-level domains? That was written up in a report, and based on that report, we gave advice to the Board to carry on and do that, but please do it a piece at a time.

The root server operators, the organizations that operate these servers are represented inside RSSAC, but RSSAC does not involve itself in operational matters. So this means two things. One is that RSSAC is not a tool to tell the root server operators what to do. It produces advice which can affect root server operators, and that advice is very likely to be well received by the root server operators, because the root server operators are inside RSSAC and help produce that advice. So it's already well-discussed between the root server operators when it ends up being a formal advice. So,

usually taken on board very gladly, but it's not a formal way to tell the root server operators what to do.

The other thing is that RSSAC is not representing the root server operators. This is not a group or collaboration or something where the root server operators have a one single voice. The root operators are each independent from each other and independent from ICANN. So, in order to have a statement from the root server operators, it needs to carry 12 signatures.

RSSAC Organization. Where is the main committee? It's composed of primary representatives from each root server operator. That means 12 people. I'm one of them. Ken is one of them. And Jeff at the end of the table is one of them. It also contains alternate representatives from each root server operators. That's another 12 people. So who can step in in case when the primary representative is not there to be able to vote or take part in discussions? We have a set of incoming liaisons from other bodies. We do interact a lot with other bodies inside ICANN, and we are also outside ICANN, actually, and we are happy to maintain this communication. We really want to interact with other bodies because we need to understand what other parts of ICANN and the Internet community expect from the root server system, and we need to be able to convey what we are able to produce and return to the community.

But that's about it. So that's around 24+ some, so just under 30 people, and that's a little too few to carry out investigations, long discussions, writing documents, and whatnot. So in order to help

us in the committee, we have the RSSAC Caucus, which is a body of volunteer subject matter experts from all around the Internet. These members, I believe the number is around 100, and they help us to do a lot of very important work. To become a member of the Caucus, you have to go through an application process where we try to find out if you have the right qualifications to be able to help us with the work, and you do that by submitting a Statement of Interest. And you also have to be public about this. Giving this Statement of Interest will become a public document.

Right now, RSSAC's chair, Jeff, is at the end of the table, and we have our incoming vice chair. I think I saw him. At the very far corner over there. Ken is our outgoing vice chair, and Hans Petter is our incoming vice chair.

We also have a number of incoming liaisons, as I said, from other parts. We have a liaison from the IANA Functions operator from PTI, the part of ICANN that operates the IANA Function. We have a liaison from Verisign who is the root zone maintainer, which is a very important step in the production of the root zone. We have a liaison from the Internet Engineering Task Force, the IETF, which is appointed by the Internet Architecture Board, which is the highest technical guidance body on the Internet. And we have a liaison from ICANN's Security and Stability Advisory Committee. We work very closely with them because we often have items or issues that touch both sides, so to speak.

We participate in ICANN work in other ways, too. We have outgoing liaisons. We have one on the ICANN Board, and that is Wes who I believe is not in the room right now. He left? Yes. We have one on the ICANN Nominating Committee. We have one on the Customer Standing Committee, which is a very small committee. That's Hirota at the end of the table, yes, who's a small committee that oversees IANA's work on the naming side. IANA has many different tasks, but when it comes to handling these requests for changes in the root zone, that's the number of procedures where the Customer Standing Committee oversees that and makes sure that it all works well. I've been on that committee, and even chaired it for several years, and I can tell you that on this Internet, the IANA is the best functioning part of it all. I am so amazed in the work they do, it's really stellar.

The last one is another one of ICANN's small committees, the Root Zone Evolution Review Committee, the RZERC, where I happen to be the liaison right now. And we look at technical modifications to the root zone. Not adding and removing top-level domains, a regular daily work we don't bother with because that doesn't change the technical qualities of the zone file. But if we were to add new types of data, new types of information, for instance, RSSAC could have been involved when we added secure DNS things to the root zone. That was a technical game changer, and that would have been addressed by the Root Zone Evolution Review Committee if it had been around at the point. It's a fairly new committee, so it

wasn't around. But that's the type of things that are handled in that committee.

UNIDENTIFIED MALE: I just wanted to point out in your previous slide, you mentioned an incoming vice chair and an outgoing vice chair. The incoming and outgoing in that previous slide are totally different.

LARS-JOHAN LIMAN: Ah, yes. You're right. Sorry about that. Probably for my lacking English.

The Caucus, as I mentioned, is comprised of over 100 DNS experts. Most of the work is conducted on video calls and exchange of e-mail, but we do meet with the Caucus. Is it twice a year, or even three times a year? Twice a year, I think, altering between ICANN meetings. So there was a Caucus meeting just today, earlier, and also at the IETF meetings, because many of the Caucus members are also active in the IETF. As I said, you have to make a public Statement of Interest to join, and in return, you get public credit for the work that you contribute to the Caucus. Again, in order to widen the set of expertise that we have at our disposal to help us with various work tasks, and it's also about transparency about who does the work, all documents produced by the Caucus will have a list of the participating Caucus members listed in there, so that there is transparency about who did what, and it's our framework for getting work done.

I'm going to go into another part here, which is a little about the principles to begin with. There is an ongoing working group, which is not the RSSAC, it's called the Root Server System Governance Working Group, which I'm going to get to very soon. We realized, when we started to work around this, that we need to have a set of principles for how the root server service is supposed to be carried out.

So the root server operators and the rest of RSSAC sat together and actually thought about this. There was quite long deliberations, and we ended up with these 11 principles for how the root server system is operated. Some are quite natural. Some probably require a bit of explanation. So, we would fully believe that the Internet as we have, it wouldn't be possible to have it if you didn't have a globally unique public namespace. So that's a statement made by the Internet Architecture Board in a specific document, and this principle here is saying that we support that document. We support the Internet Architecture Board in its stance. This is of ultimate importance for the operation of the Internet as a whole. Without that, it will simply break down.

IANA is the source of DNS root data. I'll get back into that. We must be stable and reliable. Diversity. If we make architectural changes to the system, that should be a result of technical evolution and technical needs. We rely on the IETF to define the protocol for the DNS. We must operate with integrity. We must be transparent. We must collaborate within ourselves, but also with the rest of the

community. And we must be autonomous, independent, and neutral and impartial.

Of these, I would like to highlight a few. The first one is Principle 2. The IANA is the source of the DNS root data. Every instance of the root server system, all these, over 1900 and almost 2000 machines, have the exact same data in them. There is no difference. Regardless of where it's located, regardless of which operator who operates it, they all have identical databases. So it does not matter which root server instance you query. Which letter doesn't matter, which IP address belonging to that letter doesn't matter. IPv4, IPv6 doesn't matter. Always you will get the same answer for a specific question.

The data originates from the IANA and goes through a process that can describe to the root server operators. So, IANA is the source of this information. And this principle is stated extremely clearly by all the root server operators. We take our data from the IANA, nowhere else.

The DNS is a hierarchy with a single, global, unique root. Again, we need one namespace. We cannot have multiple because that will break the system down. And all clients of the root server systems are treated equally. We do not prefer questions from some specific part of the Internet, some IP address or something. When the question or the queries come into our servers, we respond to them in order as they come in, and we respond with the same

information. So it's very equal for all incoming queries, and thereby all clients.

The diversity of the system is a very important strength. We are different. The root server operators are different in so many ways as we can invent and come up with. If we find a way that we are alike, we look at that and say, "Can we be diverse? Can we change here? Can we make change that we are not alike everyone?" Because being alike is a vulnerability, and if that's hit, it can take out more and bigger parts of the system than we would like. So there are very, very, very few things that we do in the same way. One of them is that we operate the same data, but we cannot get away from that.

So the diversity in, again, hardware, software, operational modes, types of organizations, locality on the globe, whether we are a Swedish organization, you're an American organization, the RIPE NCC is a Dutch organization, all these things contribute to the diversity, but we have the same goal, same data.

We must collaborate and engage with our stakeholder community. We do that as much as we can. We try to be reachable. There are ways to come in contact with the root server operators, if you would like to. We are here at the ICANN meetings. Please come and talk to us. We are not dangerous. I promise to put my fangs away when I talk to nice people. So we actually love to hear your questions. We love to interact with people. We do see that there are lots of myths and misconceptions, and the best way for us to help

fix that is to talk to people and inform and also listen to requests and requirements.

We also have to be autonomous and independent because we don't want any specific organizations or forces or governments or anything to have an impact on the root service. So we have to be independent from as much as we can. Actually, from ICANN, it needs from each other so that we don't have cross-dependencies between the organizations and whatnot. So that's another thing that we are very careful with.

This Root Server System Governance Working Group came from some work that RSSAC did and published in 2018. The document is called RSSAC037. It's a proposal to evolve the government's model for the root server system and improve its accountability. Since we are so independent from each other, it's difficult for us to produce things that are coherent in a way so that people can access it in a good way. We work on that. We try to do it as good as we can. But there are a number of things where accountability can be improved, and we would like to create a system around that. There hasn't been much of a governance system for the root servers in the past 30 years because the old system literally died, and we're talking about a human being that died. And it's a very difficult thing to try to modify because it's very unclear who has the authority to make changes here.

So, from that proposal, the ICANN board of directors created the GWG, and we are working from these 11 guiding principles, trying

to create a model for a new governance system. That's ongoing work. We have a very first draft of that, it was out for public comment, and that closed just a couple of weeks ago. So we are now digesting these public comments in order to improve on and balance the model a bit.

So the Governance Working Group is comprised by one member from each root server operator, two members from ccNSO, two members from the Registry Stakeholder Group here inside ICANN, two members appointed by the Internet Architecture Board, so they come from the technical side, two liaisons from the Board, one liaison from the Root Zone Maintainer, meaning Verisign, and one liaison from IANA.

Just quickly, a few points. If you have a relation to the root server system, we would like you to—this is more DNS than the root server system, but please use a validating resolver if you have one—do participate in DNS technical communities because that's where a lot of talk about DNS happens. Here it's mentioned NANOG, APRICOT, RIPE in different regions. DNS OARC isn't on there, it should be. RSSAC Caucus, please come and join us. And if you're interested in hosting one of these anycast instances, come and talk to us. There are plenty of root server operator members here. We are happy to talk to you or send an e-mail to the ask-rssac mailing list. Here are some additional resources if you want to listen to them. Thank you.

YAZID AKANHO: Thank you so much, Rob, Ken, and Lars. That was wonderful. We still have five minutes, and I guess people have questions, probably. So the floor is open. We have up to 30 participants remotely. So if anyone would like to ask a question, feel free to raise your hands or open up your mic. Thank you. Angela?

ANGELA: Hi, everyone. My name is Angela. I am with .pw ccTLD. Thank you so much for the presentation. It's been very enlightening. So I've got a couple of questions. I'd like to know what happens on your end when a ccTLD is retired?

Then the other question is, I often see that most authoritative nameservers are configured to be recursive, so what is the issue with that apart from amplification attacks and all that?

Perhaps the last one, so I can give others a chance, someone mentioned that the physical people that get to open the vaults during signing ceremony, and during the pandemic, I think this could have potentially been an issue if the people are geographically spaced. So were there any lessons learned if this entirely affected you, and if there's something else you can think about considering the natural disasters that are happening? Thanks.

KENNETH RENARD: Thanks. Good questions. I went back to the slide because at least two of those questions, the TLD, that's on the left side. When it gets

retired, when anything changes, that's between the TLD operator and IANA. So that's outside of the root server system. From our perspective, we get a updated copy of the zone today, we get maybe two or three a day, and whatever's in it, we serve it, we validate it, make sure security checks, things like that.

The same thing with—you're talking about the key signing, that's at the root zone maintainer there, and so that's, again, outside of our domain. We will see a change in the root zone, but we see those changes a couple times a day anyway, really no effect on us.

LARS-JOHAN LIMAN And then you asked about recursive service. I would argue that configuring a DNS server as both recursive and authoritative is no longer best practice. When I started 35 years ago, that was the only type of servers we had. But we learned along the way, and we realized the problems with that. The big problem with having one which is both recursive and authoritative is that the recursive one has this cache which is filled with more and more information as more and more queries come in, and that will influence the responses. So the server will mix the cache data with the authoritative data that it has in its zone files, so it may be polluted in a way by that information. And by keeping the servers apart, we can have the cache which has possibly strange data in it on one side, but the authoritative server will only rely on what's in the files and the database and respond with that. So it's a cleaner design.

So if you see that they are both somewhere, please recommend them to split that aside.

Signing ceremony? No, that's also outside the root server system. But yes, it was a challenge during the pandemic. There were, I wouldn't say, extreme measures taken in order to keep things running, but it was well done and we are now back to normal operations again. And I am quite sure that the IANA, who is responsible for these procedures, have learned a lot from that, but please talk to IANA about that.

YAZID AKANHO: We have Anupam who raised his hand. But, Anupam, before I hand over to you, let's have—oh, okay. Anupam, over to you, please.

ANUPAM GAUTAM: Thanks. My question is that recently SSAC has published a document which says that there is a lot of open source and DNS. So just wondering whether RSSAC has an opinion on the subject.

ROBERT CAROLINA: Hi. I'll take a swing at that. Just to make sure there's no confusion about this, the company I work for, ISC, is a maintainer of one of the open-source DNS software packages, Bind 9, and when I say one of, I mean the world's most popular DNS implementation. Yes, we're a little proud of that. That's true. But there are other fine open-

source products out there that you can get for exactly the same price as ours, which is to say they're free.

What's our view of it as an operator? We view open-source software as a benefit, not a burden. In fact, the one of the original purposes that our company was set up was because network operators realized in the early days that there wasn't necessarily a strong profit motive to create DNS software. It's a very, very niched product. Now, there are proprietary products out there, and many of them are very good. And if people want to pay for them, that's fine. And I believe that some of the root server operators use proprietary packages as well. We are agnostic on the question of which types of software we use, and across the entire estate of 12 operators, multiple different software packages are in use. And many of us use more than one piece of DNS software because even we don't want to be caught in the wrong place if something goes wrong with one. So even as a maintainer of DNS software, we use DNS software from other people because we can't afford for a failure of one piece of software to stop our ability to serve. I see somebody who's very, very keen to follow-up, the mysterious gentleman at the end of the desk there.

YAZID AKANHO Hans Petter Holen?

HANS PETTER HOLEN: Hans Petter Holen, CEO of RIPE NCC and operator of K-Root. I think, from our perspective, open source is great because it gives everybody the ability to use and review the software. However, as an operator, it's important for us to have a support contract with the key software that we're using, and I think that's something to bear in mind for all companies that can afford it, so that there is actually somebody that pays the programmers of the open-source developers, whether it's a big company or it's independent. I'm just making a pitch, therefore, the moment you get rich, then contribute something back. And if you're not rich, then volunteer for an open-source project by contributing security fixes and things like that. Thanks.

YAZID AKANHO: Thank you, Hans Petter Holen. We are two minutes out of the allocated time. Is there any other question or comment from the room or from the audience? Ken?

KENNETH RENARD: I'll just say that root server operators were here. There were 10 of us between online in the room. I think one had to leave. But if you have any questions, we're happy to stay after and chat with you as time runs out. But we're nerds. We love to talk about this stuff. So any questions, feel free to hang around.

YAZID AKANHO: Thank you, Ken. Thank you, Ken. Rob?

ROBERT CAROLINA

I'll just leave you with a thought. Earlier, I said that the picture I gave you where you start at the root server and go down, I told you that wasn't true most of the time. If you're wondering how much of the time it's not true, the answer it's not true 99.99% of the time. So it's only about one lookup in 10,000 where a recursive resolver needs to send a question to the root server system. That's the power of the caching data on the recursive resolver. One in 10,000.

So, in the event of what you might think of as a failure mode, think of what we're running as an airplane with 2000 nodes, 2000 engines. We calculate, we only need 5 or 10 to maintain level flight. Need more during a DDoS attack, but we've got more than enough. And even if all of them went out, which has never happened once in 40 years, we have nine nines uptime on the network, but even if they all did, the glide slope on this thing gives days or weeks or possibly months, depending on how the recursive resolver community handled it. So it's something that's so unlikely to happen is to be ridiculous, and even if it did, the impact would be virtually unnoticeable to all but a few people, and those few people would be behind the curtain, not the end user.

YAZID AKANHO

This is getting more interesting. However, unfortunately, we don't have more time. But I know the ICANN84 is still going on, and the root server operators are still around, and it will always be a

pleasure to engage with them. I know Adiel also had a light comment to make, maybe.

ADIEL AKPLOGAN: No. I just want to thank you for having this session because it is an interesting one. It helped the community hear from you directly to debunk some of the meat around the root server system, and we'll continue having that. As it say, you are around. People should get in touch with you, discuss, and we'll try to see how to engage.

YAZID AKANHO: Absolutely. Although there was no chocolate, the room was full to the end of it. So thank you again, Rob, Lars, and Ken. Thank you, everyone. The session is adjourned.

[END OF TRANSCRIPTION]