ICANN85 Mumbai | CF – How it Works: Root Server Operations
Tuesday, March 10, 2026 – 15:00 to 16:00 IST

JENNIFER BRYCE: Hello, everybody, and welcome. This session is called How It Works: Root Server Operations. My name is Jennifer, and I am a participation manager for this session. Please note that the 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 the 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 draw my attention, and I will call on you. When speaking, please state your name for the record and speak clearly at a moderate pace. I will just say there are some seats at the table. If people feel comfortable, you are very welcome to come and join us. With that, I will hand over to Ken. Thank you.

KEN RENARD: Thanks, and welcome to the ICANN Root Server System Information Session. My name is Ken Renard. I work with one of the root server operators, H Root. Today we are going to talk to you about the root server system and how it works. We encourage

questions as we go along. If you are sitting in the last two rows, you are going to be required to ask a question. So come on up if you like. We can go to the next slide, the agenda.

Today we are going to talk about the domain name system and Anycast, which ends up being a very important part of how the root server system scales and is so resilient these days. We will talk a little bit more about how the root server system looks, and then about the RSAC and ICANN Caucus, and a few things that are closer to home regarding how you can assess how you use the root server system. We can go to the next slide, actually the next two slides. To start about this, I am going to apologize upfront if this is all very familiar to you, but it is good to have a base understanding of how DNS works, and that helps us see where the root server system fits into the bigger picture.

The IP address is the fundamental identifier on the internet. We usually see them as numbers in the lower right there, but really they are strings of ones and zeros, and this is how computers address each other. This is how they talk to each other. Think of it kind of like a phone number. We do not really know anybody's phone number these days; we just click on a button. But these underlying addresses are how we talk to each other. We can go to the next slide. DNS was originally deployed in 1984, I believe. The original problem it was trying to address was that IP addresses are really hard to remember and the fact that they change. I cannot remember what I had for breakfast, let alone remember these long addresses. But the problems as the internet has grown have

evolved into the fact that IP addresses can be shared. A single web server or a single IP address may run five, ten, or a hundred different web servers that are all different DNS names. The inverse of that is that you may have a single service or a website that is available on many addresses. So which one do you choose? DNS helps address these problems.

We can go to the next slide. Here we see something very familiar, the namespace hierarchy, and it starts with a root, and then underneath that, the top-level domain, things like .uk and .org. We have different character sets as well. These top levels, then the second level, and third level follow. The main objective, or the one we think of the most when using DNS, is to map that name, something like www.example.org, into an IP address, such as the one up there, 198.51.100.52. There are many other uses of DNS besides that, for example, IPv6 addresses. This is how email systems route and find mail servers around the world. There is all kinds of other security information in the DNS that is available to protect DNS itself as well as other services on the internet. With that, we will go to the next couple slides and look up the domain name resolution process.

We are going to use in this session the diagram that is used in almost every textbook. You are probably familiar with this and have probably seen this before. It is nice and simple, and we will tell you where this diagram is either misleading or hides some of the truths that really matter to us with the root server system. It starts with an end user device. This could be your phone, your laptop, your server,

your refrigerator, or your toaster. Almost everything connected to the internet is going to take on this role as a user of the DNS. From here on the next slide, that end device is configured when it joins a network, usually with the address of one or more recursive resolvers. This is a DNS server that sits maybe in your organization's network, your ISP, or it could be external to your organization, but these are usually automatically configured. Obviously, you can override that. But this DNS resolver, and there are many hundreds, thousands, and millions of them around the world, is the workhorse of the DNS itself, so not the root server system. You are going to ask this question to this recursive resolver, which is usually pretty close to you: What is the address of www.example.com?

Next slide. This is where the textbooks usually mislead us. The textbooks say the first step is to go to the root authoritative server, that is us, the root server system, and we ask this question: What is the address of www.example.com? And we say, "I do not know, but I know where .com is." That is it. We only know about those top-level domains. We only know very little. We are not very smart. In the next step, that recursive resolver is going to get that answer and put it in its cache. From there, that recursive resolver is going to go to the .com server, which we helped them locate, and ask the same question, "What is the address of www.example.com?" And .com is going to say, "I do not know the answer, but here is where to talk to example.com." Finally, that recursive resolver is going to go to the example.com authoritative name server, and it is going to come back with the address and then give that back to the end user. This

kind of repeated pattern recursion is where this recursive resolver is going to go out and find the answer on behalf of the end user. All of this stuff will happen in milliseconds.

Where the textbooks lead us wrong is that the role of the cache in that recursive resolver is incredibly important, and it leads to some incredible efficiencies in the system. Now that stub resolver, the next time it asks for anything in the .com domain, we do not have to go back to the root server. We already know where .com lives, so we have that piece of the answer already in our cache, and that recursive resolver skips the step of going to the root server system. This keeps happening over and over again. We build up this cache, and we have lots of information in there. The other piece of this diagram that is a little bit misleading is the number of users of this recursive resolver. It is not just one person; it is hundreds, thousands, or millions of different end-user devices using this recursive resolver and that cache. That memory, and all the different things that are looked up, builds up that cache quickly and very robustly. The chances of any one query that you send to that recursive resolver ever resulting in a query to the root servers is very small. What is the estimate? .0002% or some very minuscule number of queries actually result in going to the root server. Each of these answers that I have gotten from the root, from .com, or from example.com have a TTL, a Time To Live, and that tells the recursive resolver how long it can keep this in the cache before it needs to come back and ask the question again because something might have changed. For the root, the typical TTL is 48 hours. The

recursive resolver here could go query all the entries in the root server system and never have to talk to it again for 48 hours. Again, the queries to the root are pretty rare.

We can go to the next slide. This is a little bit more about the caching and how important that is. We have that TTL. It is important to remember here that the root servers only know about the top-level domain servers and where they are. We can only tell you where to go to find information about the top-level domain servers, and from there, they tell you where to go. Again, there are incredible efficiencies by using those recursive resolvers and the cache there. We can go to the next one. That was classic DNS from the 1980s, but the system has evolved. Security, DNS security extensions, add cryptographic signatures over the DNS data. As a security person, I can never use the word eliminates risk, so I will say it reduces the risk of spoofing. That resolver can now validate those answers. That resolver will actually check those cryptographic signatures. If there is anything wrong with it—maybe somebody tried to spoof an answer, or it could just be a bit flip—it will throw away any answers that it cannot validate as being authentic.

Within the world of privacy, we saw in the diagram before that full query, www.example.com, ended up going to the root servers. The root servers do not even care about the www in example. They only care about .com because that is the only information they have. There is a standard called QName minimization, or query name minimization, that can allow us to actually preserve some privacy

so that we, the root servers, only see the last piece of it. We only see that you are asking for .com because we do not care what you are looking for. I could be going to HairClub For Men; that information does not need to be given to the root server system. Only ask the part of the question that you expect to be answered so that you are not giving away additional information. There are also privacy measures that can actually encrypt the entire transmission between not only the user and the resolver, but also the resolver and the authoritative server on the other end. As far as resilience, there are a lot of measures that we take to have multiple servers that have the same IP address, and we will talk about that on the next slide.

Anycast is a very important piece of how we deploy the root server system that makes it very robust. We will talk about unicast first. This is one source and one destination. This is like a phone call; my phone is talking directly to your phone, and that is it. There are no other copies of your phone, hopefully. With Anycast, there end up being multiple places that all share the same address that all serve this. You can think of it like going to Starbucks. I can go to any Starbucks; it does not matter. I am just going to go to the closest one. When I send a packet, a DNS request from my computer or from the resolver to an authoritative server, the internet routing is going to decide which of the many instances it will go to. It will go to exactly one. We will see this pictorially in a second, but it allows us to have multiple instances where we can have a lot more resilience. Next slide.

Here is a quick net diagram of unicast: single source, single destination, and we see the path. Essentially, those circles there are routers or hops on the route. One source to one destination, and that is the path that it is going to take. There are multiple paths to get there, but the key point here is there is one source and one destination. Next. With Anycast, now I have three different locations. If my destination there on the bottom goes away—maybe I take it down or I am doing maintenance on the machine—the routing on the internet will automatically take care of sending those queries to any of those other destinations. This is something we do that allows the 13 root server identifiers to actually have over two thousand network locations. When you hear the root server system has 13 root servers, it is actually over two thousand by using this Anycast. One more. Another benefit of this is in a scenario of a denial-of-service attacker; in the very wide scope of the internet, a denial-of-service attacker is going to have some very localized source, and the bulk of the traffic is going to go to one or a small number of Anycast sites, leaving the other sites and a lot of the rest of the internet unaffected by it. That is Anycast. I will turn it over to Liman for the rest if there are any questions we want to take at this point.

LARS-JOHAN LIMAN Thanks, Ken. I will continue from here. My name is Lars-Johan Liman. I work for NetNod, another one of the root server operator organizations. In my case, I am based in Stockholm in Sweden. Regarding the root server system today, we will start by mentioning

a few important names and terms here so that I have something to talk about. You might have heard about the Internet Assigned Numbers Authority, or the IANA. This is a part of the ICANN organization which does administrative and technical work regarding, among other things, the root zone. In the root zone context, it is the top-level domain and root zone part that is important with the IANA. The root zone in turn is the database that the root servers contain, that they hold and from which they give out responses. This is a limited set of data, and it is actually fairly small. When I gave presentations like this twenty years ago, I could say you could fit it on a floppy diskette. Not anymore, and floppy diskettes do not exist anymore, but it is still a very small amount of data, and it is actually publicly available. You can download it using your web browser. You can look at it, and that is not fake. That is the actual zone file that we use as root server operators. That is the information, the data that the root server holds.

The root server system is the entire system of all these 2,015 servers today. This is a number that changes all the time; usually, it climbs slowly. All these servers together form one contiguous system, and you can talk to any of these 2,015 servers, and you will get exactly the same information from that root zone that I just mentioned. We also have the root server operators. These are 12 organizations that manage this entire fleet of 2,015 machines around the world. They are very different in their organizational types and their locations, and we try to be as diverse as we can in many respects. We even have a motto that says, "Diversity is good," because we want to

avoid any single point of failure. If something breaks for one root server operator, another operator should be different and therefore does not contain that part that breaks, whether it is a logical way of thinking, a legal problem in the country, or the machine that breaks down. We try to make everything different except the root zone. We serve exactly the same data.

We also talk about root server instances or root server Anycast instances. These are the actual machines, these 2,015 machines across the entire globe that provide this service. We also have the Root Server System Advisory Committee, or RSAC. That is the committee inside the ICANN organization that is responsible for giving advice to the ICANN board and to the ICANN community about things that relate to the root server system, and all root server operators are represented inside RSAC. I will talk more about that. There are 13 identifiers. They are denoted by letters, and each of these identifiers has an IP address, or actually two IP addresses, one for IPv4 and one for IPv6. You can see these letters, the IP addresses, and the name of the organization that is responsible for operating that identifier. You can see that there is a diversity there. We have corporations, universities, government agencies, and membership organizations. ICANN themselves operate one of them. There is a great deal of diversity there. Many of these usually attend ICANN conferences as well, and we are happy to talk to you. We like to talk to you, so really do not hesitate to walk up to us and say, "Hey, I have this question about the root system. What is this,

or why is it done this way?" We really want to engage with the community.

MOHIT

I think you mentioned that these are diverse. I notice that all of these are based in the US. Is that a risk because these are concentrated, because again, the internet is global? What is the implication of this if all of them are based in the US?

LARS-JOHAN LIMAN

Thank you. There is a listing. There is a heavy side on the US side, yes, but not all of them are based in the US. Three are not, and these are: I Root, NetNod, based in Sweden; K Root, the RIPE NCC, based in the Netherlands in Amsterdam; and M Root, the WIDE Project, which is based in Japan. But I grant you that there is a very heavy side in the US, and we take comfort in that we have a couple of them outside the US. Historically, the system has grown over the years. It started back in 1984, as Ken mentioned, with four servers, because in 1984, the internet was something you could probably fit in my pocket. It has gradually grown over the years. As of 1998, we have 13 identifiers. Back then, it was 13 physical machines, and it wasn't until 2001 that we started to have more than 13 physical machines on the network. But then we realized we must do something, and Anycast was invented, and we said, "Yeah, this is the thing. This is for us." We jumped on that bandwagon. The number here says more than 1,900, and yes, it is over 2,000 now.

The root server operators maintain a webpage, root-servers.org. Note that this is different from root-servers.net. Root-servers.net is used for the actual service. That is the important name that we operate the service under. We use root-servers.org because that is not as crucial. We do not need to keep this system up to the same level of redundancy as the actual service. This is just a webpage illustrating what is going on and where we as root server operators can reach out and give information. This webpage has a clickable map where you can click and see where there are instances deployed, where they are located, and which letter, and thereby which organization, is operating that server. Please click away.

UNIDENTIFIED SPEAKER I guess this is an older map. It doesn't show any in India, but I think India has a bunch of instances.

LARS-JOHAN LIMAN You are quite right. This is an older map. You see nearly 1,900 instances, so it is probably a year or two old, but it looks virtually the same. The numbers are different today, so please go and look for yourself. Yes, sir.

UNIDENTIFIED SPEAKER Thank you. Just to clarify, the instances are all run by one of the 13 root server operators?

LARS-JOHAN LIMAN

Yes. Each instance is attached to one of the 13. That is right. We operate a server in Singapore, and someone else operates a server in Kuala Lumpur. Often we have servers in the same location to create redundancy at that location as well. We can both have a server in Johannesburg, maybe; I am just pulling names out of a hat. This is also a rather modern diagram showing the number of queries. We respond to a load of queries. You can see that the peak there is 150 billion queries per day. These 2,000 servers have some work to do, but on the other hand, they can handle so much more than that. The system is massively over-provisioned. We can handle many times that load that you see on the graphs. We can handle various types of overload attacks and have so far, knock on wood, been able to cope with the attacks quite well. When did you last notice that the root server system didn't work? I hope it will stay that way.

There was someone who looked like they wanted to ask a question. Maybe not. There is an interesting dip in 2021, and I believe that to be due to a reconfiguration of the Chrome web browser, which used to do a couple of interesting DNS queries to find out whether it was located on the public internet or locked in a hotel network or a private network somewhere. It did so by sending queries to the root servers. Every time someone started a Chrome web browser, we would get some queries, and you can see that that was actually a significant number of queries. We talked to Google and explained the problem, and they looked at it and fixed the code. When they

released the new version of Chrome, you can see that the queries dropped.

There are a good number of myths around the root server system, and I will try to dispel a few of them. There is a myth that the root server system controls the internet traffic, and they do not at all. These two thousand servers couldn't handle even the smallest fraction of all the web traffic and all the video browsing. That is so much more, and the root servers were never intended to do that. You talk to them to understand where the video server is that you need to talk to, or where the websites are that you want to talk to, or to where you shall send your email message. This is just guidance to which server to talk to. The pieces of equipment that really handle the traffic on the internet are called routers, and these are pieces of network equipment that build up the internet service providers. You have a small router at home, usually a Wi-Fi router or a small home router. The closer you get to the central parts of the internet, the bigger these routers get.

The next myth is that most DNS queries are handled by a root name server, but Ken already dispelled that myth by showing how it actually works and that the caches carry the major load of DNS queries. As mentioned, the cache builds up very quickly, and once the cache is there, most queries are responded to from the cache because it already has the information. It can reuse the same information many times until the Time To Live ends, and it will have to go and refresh the data. It needs to go back to the authoritative server and ask for a new copy, and it receives the information with

a fresh Time To Live and can reuse it again. Administration of the root zone is handled by the root server operators. No, that is not true. The administration is handled by the IANA. We do not do anything with the data. We receive it, we put it on our servers, and we leave it there so that the entire internet can query for it and get the information. We are even unable to change it because it would be immediately noted if we tried, because of the security signatures that Ken mentioned. We do not have access to these security keys. That is purely the root zone maintainer, which sits in the path, who has these keys. We do not. If we try to change something, these signatures will no longer match, and your resolver will immediately see that something fishy is going on. It will not touch that data and will just leave it.

Next, the root server identifiers have special meaning. It used to be believed that a.root-servers.net had a special meaning separate from b.root-servers.net, and that is definitely not the case. They are all equal. It is not the case that anyone has information before anyone else or is responding quicker or in any other way. They are actually all identical, and we have these letters only in order to distribute this traffic over these 26 identifiers or 26 IP addresses. In total, there are more than 13 root servers; that used to be true, but that was 25 years ago. We are over 2,000 today. The root server operators conduct operations independently. Well, we do operate independently to some degree, but we collaborate a lot because the root server system as a whole, all these 2,000 machines, need to respond in the same way. That means that we need to

collaborate to make sure that happens. We have frequent meetings; we meet three times per year. We do not meet at the ICANN meetings; we meet at the IETF, the standards organizations meetings instead, because of old traditions. We discuss technical issues, security issues, and how we collaborate. We also have collaborative tools. I can make a phone call from here and now, and the phones will ring at fifty places across the globe to say we need an immediate telephone conference now because we have had a major problem. We have not had to use it for any major problem, but we have done it as an exercise at every meeting to make sure it works. We have email lists and video conferences so we can reach out to each other. Thanks to these frequent meetings, we also know each other quite well. If Ken calls me, I recognize Ken's voice because we have been to so many conferences over the last 25 years.

Root server instances receive the entire DNS query. That could be true in the past as well, but as Ken mentioned, we have modern standards that limit the content of the query, so the resolver will see the full query from the stub resolver. The phone or your laptop will send a full query to the stub resolver nearby, but the stub resolver will usually today send only the absolutely necessary part of the domain name to the root name server, which is usually just the name of the top-level domain that it is looking for.

JENNIFER BRYCE

There's a question. I'm going to read aloud a question that's in the chat here. If most DNS queries are handled without the need for a root server, and there are now more than 1,900 instances globally, not just 13, what does this tell us about the actual role and workload distribution of the root server system in everyday internet operations?

LARS-JOHAN LIMAN

Thank you. The load on each server varies a lot. It depends on where it sits and how many resolvers are around it, what the clients around each resolver ask for, and how many queries can be answered directly from the cache. The fact that we have so many servers deployed is for redundancy. We really want to make sure that the service is available everywhere on the globe as far as we can. There are limiting factors in what we as root server operators can do, but we really try very hard to make sure that it is very good everywhere. To deploy in a geographic location limits the amount of time a DNS query has to travel across the network to reach a root name server and get a response, so you can get responses quicker if the root server is close to the resolver. It also helps to suck up traffic in attack scenarios. If someone is trying to overload the system, having many servers helps to sync that traffic and be able to cope with it without being overloaded. These are the main reasons for having this many servers.

I agree the importance of the root servers is slowly declining, partly because other mechanisms are being used for finding things on the

network. When did you last type a full domain name into your browser? No, you Google for something interesting, or you use AI to find something interesting, and there are links that you click on. That kind of slowly, in my personal view, removes the dependency on the domain name system as an identifier system. It is going to be there for a long time to come, well into my retirement, but I think there is a gradual decline in the dependency. Another reason is that it is quite possible these days to operate a resolver without ever talking to a root name server for DNS resolution by downloading the root zone. I said it is publicly available, and that is true. You can download it into your resolver, so the resolver already has all the root zone information. It is not much data, and it can use that information without talking to a root name server ever. Then you have it very nearby. But you need to have a mechanism to update it. The root zone is updated two times per day, and you need to keep your own copy updated accordingly. There are documents that describe how to do this; it is referred to as hyperlocal root. Come and talk to me if you want pointers for how to do that.

JENNIFER BRYCE May I insert a few more questions? There are a couple here. First of all, given that root server operators coordinate their operations as a collective rather than independently, and IANA, not the RSOs, handles administration of the root zone, how does this division of responsibilities contribute to the stability and security of the DNS infrastructure?

KEN RENARD

This is Ken. We will see this in another slide or two, but the division of responsibility is on purpose. We do not want any one organization to have control over the entire system or have the ability to make changes on their own. IANA controls the contents. In fact, there is another step: the RZM, Root Zone Maintainer, does the cryptography and the signatures. We are just the IT department in the basement that runs the computers. It is simple, but that division of responsibility is a security feature designed into the system to prevent malicious actors from having too much control.

JENNIFER BRYCE

Thank you. Another question. Do these RSOs monetize their business by access to the root servers? If not, how do they keep maintenance of all these instances of root servers?

LARS-JOHAN LIMAN

That is a very good and obvious question. The root server operators are not paid for the root service from anyone. ICANN does not pay for this service, nor does anyone else. All the root server operators do this for the good of the internet. This is a pro bono service that we provide. That said, running two thousand servers across the globe is not cheap, so it costs money, and it is down to each root server operator to find a way to finance this somehow. Here is where we also have a very large degree of diversity. The different root server operators finance their service in totally different ways.

I can speak for NetNode. We deploy servers across the globe, and when we do so, we install a platform which can do more than operate only the root. We can provide service for other DNS purposes, which we can sell, and we use money from that commercial business to pay for the root service. That is our way of doing it. You can look at the RIPE NCC, for instance. They are responsible for allocating and assigning IP addresses in the European area. They are a membership organization; each member pays a membership fee, and from that fee, they take some of the money to operate K Root. Ken, do you have a different way?

KEN RENARD

I just want to address one other part of the question. We do not monetize the data. We consider the data private. We use the data only to support the service and keep it running efficiently. That is it. We do not sell the data or release it to anyone for monetary purposes at all. Part of our core principles is that we will never do that.

LARS-JOHAN LIMAN

Very good point. No root server operator sells data. We do contribute to research efforts here and there, but that is always under very strict non-disclosure agreements. When we deliver data, we often anonymize it, so you can see what the query was about, but you cannot see the IP address that sent this. That can be used to look at the system and make it more efficient and understand things like name collisions inside ICANN. It was

important to understand the content of the traffic in order to understand the risks of name collisions.

UNIDENTIFIED SPEAKER I think for the audience, one obvious question would also come in is, why limit to 13? That would be worth getting into as well.

LARS-JOHAN LIMAN Why are there 13 root server identifiers? That is due to an old limitation in the domain name system, where you wanted to fit the list of all root servers and their IP addresses into a single packet on the network to be efficient. Since then, the network and the DNS protocol have evolved, and today you can do it in a different way. But since this evolution started, there has not been a process to replace a root server operator. No one knows how to add or even remove a root server operator. I cannot quit; I am stuck here. To address that problem, the Root Server System Advisory Committee first did a problem inventory. This is an ongoing long-term project. The RSAC first did a problem inventory and said, "We would like to change the governance system around the root servers to make it evolve with time and make it more transparent." That led to a cross-community working group here in ICANN called the Root Server System Governance Working Group, which just a week or two ago sent its final report to the ICANN board with recommendations about how to proceed and create new structures around the root server operators. Hopefully, there will be a process in the future for adding, removing, or changing things.

But there is also no real reason to make a change. The system works quite well as it is. Since 2001, we actually have not 99.9999% uptime, but 100.0000% uptime, not for each individual server, but for the system as a whole. Knock on wood.

KEN RENARD: Another point on the "Why 13": we saw in the history document that as the internet grew in the eighties and nineties, we needed to add more. But with the advent of Anycast, we can grow in an entirely different dimension. Each one of those 13 can now grow without bounds. That 13 is now over two thousand. The technical infrastructure is sufficient. The need to go beyond 13 is what exactly? Thanks.

UNIDENTIFIED SPEAKER: No more questions for now. Oh, there are some in the room.

LARS-JOHAN LIMAN: Please.

SUNCICA ROSIC: Suncica for the record. I have a question about Anycast and packet fragmentation. I definitely understand the importance of Anycast for internet resilience, but what happens in cases where we have an IP address that consists of a lot of packets? Can it happen that those packets are sent to a different Anycast node? If so, how is it

handled, especially under UDP where speed matters quite a bit? Thank you.

LARS-JOHAN LIMAN

Now you are diving deep into the technical details. I will be happy to answer. You are right; it can happen that these packets end up at different Anycast nodes. It is possible, but it is rather unusual. It happens when you have either a load-sharing system where traffic is shared on separate links, and if you are unlucky, it can take different routes at the other end of these parallel links. It can also happen when you have what is called flaps in the routing system. When the routing system on the internet changes, if you are unlucky, your two packets are crossing just between a flap, and they can go in different directions. As long as you are talking about single UDP queries, it usually works out well. But if you have fragmented packets, the two fragments can happen to go to different servers, and then you will not receive a response because neither of the packets is complete. The server is unable to generate a response from that because it is unable to read the full content. There is an entire problem space around that, but overall it is a minute problem. Theoretically, this can happen, but in practice, it turns out that it works actually much better than I expected twenty years ago when I turned on my first Anycast node. If you want more details, come and talk to me afterwards. Yes, please.

UNIDENTIFIED SPEAKER: There were a couple of DDoS attacks on root servers in 2015, and from what I know, nothing serious happened. It was a very minor issue which was resolved quickly. You said that currently there is almost 100% uptime. It has been 10 years since then. Do you foresee any theoretical problems that could arise, given new technologies which have not been utilized? What are the potential vulnerabilities that you think theoretically exist which could be used nefariously?

LARS-JOHAN LIMAN: That is an open minefield. This goes into unknowns—the things we do not know. The root server operators are involved in very many places. You will find us here at ICANN meetings. You will find us at IETF meetings involved in the development of the DNS protocols and the BGP protocols for routing. You will find us at the OARC meetings. OARC is an association for DNS-related people and operators. We try to stay on top of what is going on and what threats there are. I am a member of at least two security-related organizations inside Sweden, with the ISPs, for physical security and cybersecurity. We are involved in as many places as we have resources to be. That said, actual threats are always out there, and they come in new ways every day. I have no idea what AI is going to do to this. We can just try to stay ahead of the curve and be prepared for what is coming. The systems are really over-provisioned, but if there are logical ways to attack a DNS server that we do not know about—where you trick the server into doing strange things or you find a very nasty bug in all implementations

of DNS servers—then yes, things can happen. That is part of operational life. Please.

UNIDENTIFIED SPEAKER: Thank you. On the risk assessment, and this time not the cyber risk: you explained very well how diverse the revenue making was for each organization to be able to finance this. Do you see a risk with the potential increase of hosting, memory, and the impact of AI, not on the cybersecurity thing, but on the fact that it makes the price higher? The second question is, in the discussion we have today about the governance of the root server, is there something about trying to have a more predictable revenue for the root operators in case there is a huge problem of price rising?

LARS-JOHAN LIMAN: Thank you. I do understand the question. The first one regarding price issues is starting to show. We are somewhat limiting ourselves in how many servers we buy and deploy, but we are also at a very good point. I firmly believe that this price increase will be temporary because there is money to be earned by producing these things. The people who produce memory chips and other scarce products will increase their production, or someone else will start to compete. In two or three years from now, I am quite sure that the prices will start to go down again. We are in such a good place that if I stopped buying servers now and didn't buy any until

three years from now, we would still be quite fine. It would not be a problem.

UNIDENTIFIED SPEAKER I just want to interject with a timekeeping note. We have five minutes left.

LARS-JOHAN LIMAN I was just about to say that, and we need to try to finalize the slide deck. Regarding root zone versus root server system: the root zone is the actual data. That is what we receive from the root zone maintainer—this database, this lump of information. This is the list of top-level domains with the information for each top-level domain and which servers are responsible for that TLD. This is managed by ICANN through the IANA and is administrated according to community policy. The policy is set here at ICANN and then implemented by the IANA, and then we have the distribution system that eventually hits the root servers. The root server system is the system of all these servers, and it is implemented via these 26 addresses. It is a purely technical role. We do nothing with the administration of the data. We do not change anything, add anything, or take anything out of it. The root server system is the responsibility of these 12 organizations that operate this service.

This is the schema for how the root zone is pushed through the system. Changes to the root zone are initiated typically by the top-level domain operators. If the Swedish top-level domain wants to

make a change to the set of servers that serves .se, they would contact ICANN. There is a web portal for doing that, and they would ask to have that changed. The IANA will review the request, make the change, and send this change information to the root zone maintainer, which will then sign the information with crypto signatures. The output of that is the root zone, which is published and distributed to the root server operators. The 12 operators pick up this zone file from the root zone maintainer and then distribute it to all the server instances. In our case, we are responsible for roughly one hundred of these two thousand, so we distribute it among our one hundred. Ken is responsible for whatever number he has and distributes it to his servers. We have this chain going downwards. We have, at the end, these two thousand machines that respond to the queries from the internet. All this is firmly automated from the moment that the root zone is made available by the root zone maintainer. The time it takes to have it deployed on 90% of the instances at the end is probably less than two minutes. It is fully automated all the way. The transfer of the root zone is also checked with extra signatures so that no one fiddles with the zone as it goes across the network. We have dual security there: the signatures and the transport mechanisms.

We need to mention RSAC as well. RSAC is a group inside ICANN; this is an ICANN community. From the bylaws, the role of RSAC 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 is all we do. You didn't see policy or control

operations in there. This is an advisory group; we give advice. The root server operators are represented inside RSAC, but RSAC itself does not involve itself in operational matters. Operations are handled separately by the root server operators. As I said, we meet at the IETFs; we have a collaboration there that works, but that is not an organization. It is just people that meet and work together. RSAC is composed of primary representatives from each root server operator, so there are 12 primary members. There are alternates; in case I am sick, my alternate will take over, so there are 24 people there. Then we have incoming liaisons from other bodies, such as SSAC, IANA, and other places. We are a fairly small committee and do not have the resources to do a lot of research and document writing, so we have a body of subject matter experts called the RSAC Caucus. These are people who want to contribute to RSAC's work. This is a membership where RSAC confirms you; you send an application and a CV so that we know you fit in the system, but we extremely rarely turn anyone down. It has happened once or twice.

UNIDENTIFIED SPEAKER Just to let you know, we are at the end of the time that we have allotted. If you could make a final comment, we can wrap up. Thank you.

LARS-JOHAN LIMAN Okay. I will skip through this. These are our current chair and vice chair, and the liaisons where we have them, and the caucus I just mentioned. I don't think we have anything else here; we have gone

	through most of this. There is a set of additional resources in these slides, and the slides are available as far as I know. If not, I will ask to make them available.
UNIDENTIFIED SPEAKER	They will be available on the schedule.
LARS-JOHAN LIMAN	We are here, and we are happy to talk to you. If you weren't able to ask your questions during the session, come and see me, come and see Ken, or come and see the others. We are quite happy to talk to you. Thank you very much.

[END OF TRANSCRIPTION]