ICANN85 Mumbai | CF – ccNSO: Tech Day (3 of 4)
Monday, March 9, 2026 – 15:00 to 16:00 IST

CLAUDIA RUIZ

Hello and welcome to the third session of ccNSO Tech Day. My name is Claudia Ruiz, and I along with my colleague Susie Johnson are the participation managers for this session. Please note this session is being recorded and is governed by the ICANN Community Participant Code of Conduct, the ICANN Expected Standards of Behavior, and the ICANN Community Anti-Harassment Policy.

Please observe the following guidelines to participate in this session. They will be noted in the chat for your reference. During this session, questions or comments will be read aloud if put in the proper form as noted in the chat. Interpretation for this session will include English, Spanish, and French. If you would like to speak during the session, please raise your hand in Zoom.

When called upon, virtual participants will be given permission to unmute. On-site participants will use a physical microphone to speak. Please state your name for the record and the language you will speak if speaking a language other than English. And please speak slowly to allow for accurate interpretation. Thank you. And with this, I will now hand the floor over to Luis Diego Espinoza, moderator for this session. Thank you.

LUIS DIEGO ESPINOZA: Good afternoon. Thank you, Claudia. My name is Luis Diego Espinoza, and I will be chairing in this session. Our first presentation will be by Mr. Steve Crocker, and Proposed RDAP Data Access Framework. Let's hand the presentation to Steve.

STEVE CROCKER Thank you very much. Let me make sure I get control of the slides here. Not sure. Whoops. Come on. Backwards. All right, so there's a little bit of lag in this. I want to talk to you today about what we call Project Jake. Our focus is on the disclosure and the preceding aspect of the collection of DNS registration data.

And the general picture is how can we build a more complete, comprehensive, and flexible framework so that there's a multiplicity of policies that are supported, there's greater clarity about the results, and hopefully, greater efficiency all around on all parts. So, that's the main topics. We've named the project after Jake Feinler, who was the original keeper of registration data back more than 50 years ago when the ARPANET was first getting going.

She's still around, she's an amazing lady, and she gave us permission to use her name partly for attention, partly to give her the respect. So, we seek engineering solutions to complement, support existing DNS registration data disclosure policies that have grown up in a kind of ad hoc way over many years and we've been taking a fresh look at the whole thing. And I want to show you what the basic design ideas are and how far we've progressed.

We actually have software working in partnership with the Taiwan top level domain operators TWNIC, a pilot underway. So, the key elements in this framework is that there are requesters and data holders we use, data holder as a umbrella term to encompass both registrars and registries. In some cases, requests for data go to registrars, that's common. But in other cases, they go directly to the registries, and so we have a framework that treats them both the same.

One of the key elements, so the challenge is in the world that we're living in that access to registration data has two main concerns. One is a big privacy concern and the misuse of that data that became rampant. And then the counter reaction of clamping down both individually and by governments and restricting the access to the data. Individuals clamp down by simply not providing the data or giving false data or using proxy servers or whatever.

And governments clamped down with GDPR and other privacy regimes instituted in various places. So, the question then is, can we bring some order to this in a way that allows all of the relevant parties to agree on what their policies should be in a way that has both clarity, and as I mentioned, some efficiency involved?

So, we divide the picture up into requesters and data holders, and one of the pieces of the design is that there can be agreements in advance between groups of requesters and groups of data holders that requests that meet certain conditions, don't have to be

adjudicated each and every time along the same lines, but again, under local control, under control of the people of the data.

So, we've developed a framework based on existing protocols with some extensions. Let me see if this now works. So, the picture then is that we have an operational framework with agreements between groups of requesters and groups of data holders. And in order to make those groups work, there has to be an administrative process for each of those groups. So, imagine a requester administrator and a data holder administrator. And who that is depends upon what the nature of that group is.

A group could be formed by a bunch of requesters saying, okay, we want to form a group and we'll appoint somebody to be our administrator. Or it could be formed by having some organization say, we'd like to provide this as a service and offer that and then invite people to join that. So, either way is fine from our point of view. So, what's an agreement?

An agreement contains a hierarchy of things, one is general agreement on terms and conditions, and then specific acknowledgement types of requests and a request type it says what the purpose is and what the rules are for the use of it and who's responsible for assuring that that's the way it gets used and what the accountability and mitigation or process is in the event that there's some abuse of all of that.

And then very specific levels of detail as to what level of access is being provided and what data elements are likely to be provided.

Again, all under the control of the data holder. It's an extensible framework, it's not a one-size-fits-all, it's a sort of basic part, and then the individual parts can be customized. So, from a protocol point of view, sort of think of multiple layers basically or layer one is a request and response protocol built on RDAP with extensions.

And then the next layer above is the mechanism for putting agreements in place, and then there'll be a layer above that for general sharing of experience and tools and listings, a way of advertising what is available. We're building software on the requester side. We've named the requester software JADDAR. It's a particular offering think of it comparable to bind as an offering for DNS, but if you want to build your own, all the protocols will be documented one when anybody could build it, and so forth.

We're doing a pilot as I mentioned with TW NIC colleagues here Shuping back there, Joe Fan, and Kenny Huang from TWNIC. A very, very strong partnership, very capable, smart people who see not only the technical aspects but all of the business and political aspects as well. So, it's been a quite heartwarming partnership, so to speak. So, here's what things look like. The requester side, starting from the bottom on the left, a requester software that would be used for making a request.

And here's one of the key things about this. After looking at some of the issues that have arisen over time with RDRS and other things, it became clear to us that that software needed to have a component that is persistent. That is, it's not just you make a

request and you either get an immediate response or you walk away and then you have to go back manually and resubmit the request or check it out again, but you have some process that's running on your own behalf.

A lot of things get a lot simpler when you have that idea in there. And then above that are the administrative processes for forming agreements and for having access to them. Another very important element of all of this is strong authentication of the requester. Identity is an extremely important aspect. It's not all by itself sufficient. Even if you know exactly who's making a request, that doesn't mean you can trust them or that they're going to behave properly.

So, in addition to identity, and in fact in some cases even without having the identity but simply knowing that the person who's making the request is operating in a framework with that, they are accountable. Having that information available is sufficient and provides the basis for making the necessary decisions in favor of disclosing non-public data that you would not otherwise want to disclose. So, the picture there of OpenID server using Key Cloak software at the moment, but one could use something else, provides that functionality on the requester side.

And then on the data holder side, you have an RDAP server on the bottom that is responding to requests, and it has access to agreements that are already in place. And then where it says available templates, those are the advertisement of what kinds of

agreements the data holder might be willing to engage in with the appropriate kinds of requesters. So, I'll show you some of the sort of diagrams of that.

So, there's a process for creating a requester group, and then there are process that is then triggered by that after the group is formed where an administrator for that group can add or take away members of that group. There's a similar sort of thing on the data holder side. And then the critical aspect here is that agreements can be entered into by looking at what the data holders and groups of data holders are willing to agree to, and then having a requester group asked to be included in this, filling in the details of who they are and all of their particulars.

And then that opens the door for making requests in which basic cycle is intended to be fairly simple. A requester triggers a process that involves getting a bearer token from the OpenID server that is included in the RDAP request, that's the bottom red line there, over to the data holder, who can then authenticate that bearer token and all of the other parameters, and then decide whether to respond to the request, and if so, with what data, and so forth.

Here's another diagram of the same sort of thing, the key cloak server is shown on the right there, it's the way the diagram came out, but it belongs as part of the elements on the left. Let me Insert one other very important point. It's traditionally in a lot of the discussions to divide data up into public versus private. We've listened to a lot of conversations about all of this. And the basic

distinction between public versus private data is clear enough, but in some instances, we hear things like this.

Well, if the person making the request is only allowed to have access to the public data, but there is some private data and it exists, then we won't tell them what the data is, of course, but we'll say redact it to show that it exists and that if they could find a way to get higher level access, they could get at it. Okay. But we listen even further and we find that there's other discussions in which, well, that data is, we're not going to tell you whether it's there or not.

So, we translate that into saying, well, that must mean it's more sensitive in some respects. So, instead of just two levels, there's more than two levels, and we've settled on four as a useful gradation of what you can think of as privacy levels. And then with respect to making particular requests, Part of what defines a request type is what the access level is, how high up are you allowed to go. That's not the sole criterion for what data you get back.

There's also a second dimension that some data elements might or might not be included in a request type. And you might say, well, isn't it just a question of all or nothing? One of the elements of discussions that we've listened to, again, over a long period of time, is that a lot of discussions are focused just on contact information, but there are other kinds of requests that other parties might make.

Law enforcement is interested, for example, in who paid for the registration or what was the IP address at the time of registration. There's a design choice in these things and often we see that the design choices. Those are out of scope, we won't talk about them.

We've made the opposite design choice. Everything is in scope and it's just a question of access control so that it's entirely possible to define a request type that includes, say, the IP address at time of registration that simply would not be included for other kinds of request types, even if a high level of access is provided. So, the other key thing that has emerged over time is that the assignment of sensitivity level is sometimes dependent upon frequently dependent upon whether or not it's a business versus an individual.

So, it may be normal to say if the business registration will make the physical address of the business publicly available, that's not our decision, that might be a plausible decision. And on the other hand, but if it's an individual, we won't make the physical address available publicly and we'll raise the level of sensitivity of that so that you need a higher level of access for them.

Working with Taiwan, what has emerged is five different categories of registrants, business versus individuals, but each of those is subdivided into ordinary registrations of that type or ones that require extra level of privacy for one reason or another.

Individuals, of course, sometimes I imagine celebrities or politicians may have a higher level of protection that they require than an ordinary individual even if the ordinary level of protection

is already high. Even businesses have distinctions in there. Consider for example a non-governmental organization operating in a contested area even though it's a business and there may be no personally identifiable information involved.

Providing the physical address of where they're operating may be particularly sensitive. In addition to those sort of basic two by two distinctions, Taiwan has been working with some registrars who do an extra level of verification of the Information that's provided so they validate information. So, in addition to the sensitivity of the data, a different element in the in the model is what level of accuracy or validity of the information is there.

So, we've included the notion that the data is, at least in principle and conceptually, is tagged at the time of registration as to whether or not it is taken as given with no validation, whether it meets a syntactic level of checking, whether it meets an operational level of checking. or whether there is even stronger reason to believe that that data is absolutely accurate. So, again, a four-level model there.

Taiwan has been using the term green for these extra validations, unrelated to other common uses of that color to have some ecology-oriented thing. That's not the right button. Let's try this button. So, here's a snapshot of a piece of the structure that that we have in one of our tools. This shows for the multitude of data elements that are just related to the registrant, and there's a similar set for admin and tech contact and so forth and for other elements.

This shows on the first column is whether or not that data is to be collected or whether it's optional or not collected. The middle column shows what level of validation that data gets subjected to, and the one on the right is what level of sensitivity is assigned. This is for an individual with extra privacy. This one is for a ordinary business. So, these are quite different from each other, and I'll try to flip back and forth here so you can see the difference, sorry, on the right of what the sensitivity level is.

And then if we go to the green registration and compare that to an ordinary business, the middle column shows you the contrast in the level of validation that is common throughout that. Here's an example of the data that's returned for two different accesses to the same data. The stuff on the left that's red is redacted and the actual data is given on the right for a much higher level of access.

So, there's benefits for the data holder because of streamlined access, much finer grain control of how to treat different cases and a lot more orderly operation, if you will. And taking care of a lot of the work up front about who gets access to what. And as I said, a lot of policy control. The pilot is underway, we are actively looking for others to participate.

We've been working with a couple of groups of potential requesters who not yet announced publicly, although they're not hiding anything, but they haven't reached the level of making public announcements of their participation. But in the law enforcement community and in the intellectual property community, and we

would be delighted to talk with others and have others participate with us. That's it. Thank you.

LUIS DIEGO ESPINOZA: Oh, thank you very much. We have a short on this presentation on time so we can see if at the end of the presentation we have a little bit time for more questions. Our next presenter is Jeff Osborn from ISC, Root Serve Introduction. Go ahead, Jeff. Thank you.

JEFF OSBORN: Thank you. In keeping with the tradition of being in highly technical meetings like this, I have failed to get my audio and video working, so give me one minute, that was better than I worried. Hi, appreciate your time. My name is Jeff Osborn. I'm the chair of the RSSAC, the Root Server System Advisory Committee. In my day job, I'm president of ISC, which operates a root server as well. Too close or not close enough? Closer? Okay.

And we published Bind9, which is a popular DNS software. Eberhard ran into me when I was doing a presentation. There was sort of an intro to the root server system at the last ICANN, and when offered a microphone, I would say yes. It's a little bit of an introduction, but I think it will be especially interesting because I'm kind of taking what Warren said this morning and turning it a little upside down.

So, hopefully that will lead to an interesting discussion. I would never say Warren was wrong because he'd take me out behind the

school and beat me up again, but it's a different approach. So, if I shoot through a couple of slides, it's because people here probably are more knowledgeable than the people this was designed for, who are primarily legislators and regulators who are brilliant in their own regard, but just new to DNS and the root server system.

So, forgive me if I go too fast or too slow. Introduction. The goal here is to just increase the understanding of all y'all about the root server system and the root server operators, and we're going to do that by explaining the role and purpose of DNS and the roles of the two primary players in a resolver and an authoritative server. We're going to look at how when and why a resolver consults the root server system and address a couple of misunderstandings.

So, you probably all know this. DNS is simply a way, at its simplest, to take words that humans know and change them into the numbers that the computers need. So, you know www.amazon.com, the computer needs to know 40.81.95.161. So, all connected devices basically use DNS to figure out where they need to go. The names stay the same so that the numbers can change. DNS questions are generally asked with names, and the answers generally have a numeric address. So, why? Obviously, it's human-friendly.

I'm old enough to remember having to remember long phone numbers for people, and now I just touch the button that has my wife's picture on it on the phone. Similarly, you just need to use words, you don't need to remember long addresses, and some of

us are old enough to remember having to know the HR machine was 142.13.12.19, and hilarity often ensued because you'd get it wrong.

Service portability is the other good reason. If you could take down your whole infrastructure and moved it across town and change the numbers, it wouldn't matter because the numbers will follow you, DNS will follow you to your new home. The remarkable thing about the DNS when you really look inside it is just how incredibly huge it is. This is the largest distributed database I'm aware of.

Somebody could correct me if I'm wrong, but some of the little parts of it are mind-bogglingly big. It's flexible, it works, and it's fast, and it's kind of remarkable if you spend a lot of time playing with it like many of us do. So, is this going too fast? Because I'm trying to get through the intro. There are millions and millions of resolvers out there. The obvious ones you're probably aware of would be like 1.1.1.1, which is CloudFlare, 8.8.8.8, which is Google, which I was surprised to find are the number one and number two resolvers in India.

They're also way up there on the list for the rest of the world. What the resolvers do is read the authoritative information from the internet's phone books. The phone books are things called authoritative servers, and basically an authoritative server is the authority for some set of the information you're looking for, whether it's a top-level domain, a domain name, a secondary, and

the resolvers go out and look for that information. They hang onto it, they put it in their memory, and they serve it.

So, the basic process is always the user asks, how do I get to tata.com? And the answer comes back, you get there by going to 40.81.95.116. It happens in milliseconds, and it happens about 500 trillion times a day. If you know engineers, you know how long it took to agree on that number, and I think we're probably close. So, resolvers get the addresses, and then they remember them. It's called caching.

So, eventually, a memory will time out, sometimes lightning hits, sometimes people trip over the cord. There are reasons to go back and look again, but in the normal course of business, once you've looked an address up, you have the address in cache and you return that. So, depending on how much information you need, you're going to have to go through one of four cases.

In the absolutely most common case, case number one, the resolver doesn't have to ask anybody because the resolver already looked up that information and the resolver just gives you the number. If you're looking up tata.com, amazon.com, google.com, the odds of you being the first one to look that up are nil. So, it just comes out of memory. In the second case, the domain name's authoritative server is known.

It definitely has amazon.com, but it might not have secret files from people who work at unclejeffsbookstore.amazon.com, because it's kind of a sneaky little additional word in there. So, in the second

case, you'd have to go to the domain name's authoritative server and ask it, hey, what's to the left of the other dot? In the third case, you're going to the top-level domain's authoritative server. There are about 1,450 of these out there in the world.

These are the top-level domains like .in, .com, .net, and some operator is responsible for knowing the address of every domain name attached to that top-level domain. Hopefully, that much you know about DNS or the next slides, I'm going to lose you. This is the part that has been accused of being the ugliest slide at ICANN, and I will argue that at least it works because at the end people always say, well, I understood.

The box, the gray box, is the resolver. Outside of the box is the question being asked, and the question is always the same and the answer is always the same. You're always asking, how do I get to this domain name? And the answer that comes out of the box is always, here's the IP address. But what happens inside the box depends on how much information is known and how far you have to look.

So, we're going to go through case one where we follow the numbers. Start. I'm looking for the IP address of www.example.com. Do I know the answer? I look in cache memory and there it is. So, my answer, here's the address. That's the simplest and most normal way this happens. And now if you start looking over on the right, you get some color commentary. Get it? It's red color commentary. This is what between 90% and 95% of

the resolver lookups do. All they do is they ask the resolver, you found this before, just tell me what you found the last time.

So, this is where I start disagreeing with Warren because in computer science class, they say you take a resolver out of the box, it's brand new you unwrap it and you plug it in and the first thing it does is looks for a root server. The vast majority of the machines out there have been running. They're hot, they have a hot cache, they already have looked things up. So, the idea that the first thing you have to do is ask a root server is correct only in the sense of following a logical set of steps.

In actual practice, the odds you have to go there are long, as we're going to see. So, in the next case, we'll say, okay, now I'm going to ask, do I know how to find www.example.com? No, I don't. Okay, how about example.com? Yes, I know where example.com is. So, I'm going to ask the authoritative server for example.com, I'm looking for something to the left of the dot. Do you know what it is?

And so, I asked the authoritative server, do you know where www.example.com? It does. We put it in memory, and now we ask the question, do you know the answer? And we do. We got the answer. So, the end, we did the same thing, but we needed an extra step because we had to go to another authoritative server to get more information. So, now let's say we know even less. We're asking the question, what's the address for www.example.com?

Do I know it? No. Do I know where example.com is? No. But I do know where .com is.

.com has about 170 million domain names attached to it, so when you go out and ask, you say, hey, I need to know the IP address for example.com, they look through their list, they come up with it in the .com authoritative server, and they report that address back up to memory. And so, now do you know the answer? No, you just learned where example.com is, you still need to know about the www.

So, we're going to go down and say example.com. Do we have an authoritative server? Yes, we do. What is www.example.com? Here's the answer. Put it in memory. Now do I have an answer? Yes, I do. Terrific. You'll notice over on the right we're accumulating how frequently this happens. You don't have to look anywhere about 90% of the time.

You have to go to the authoritative server for the domain name we came up with about 5% of the time. You have to go down and look at the TLD's authoritative server about 2% of the time. And if you're doing the math there, you're seeing that we're really running out of percentages. So, then we get to the final case where this is the resolver, it was just taken out of shrink wrap and plugged in, it doesn't know anything.

So, we start and say do you know where www.example.com is? No. How about example.com? No. How about .com? Ah, resolver software always contains… Slow down or shut up, slow down.

Okay. I completely lost track. Contains a hints file. So, basically, there's a very short list of the addresses for a set of root servers. And so, in this case, you need to go to one of the root servers and you say, help, where is .com?

And the root servers, which know the IP address for all 1,450 top-level domains, can tell you, here it is, this is the address for .com. And I bet you can guess what happens next, because this is a recursive process. We take that and put it in memory. We say, now do I know the whole answer? No, I don't even know where example is. I just know where .com is.

So, now I ask the .com authoritative server, hey, where's example.com? It tells me. I put it in memory. Now do I know the answer? No, but I'm getting really close. So, now I ask the authoritative server for example.com, where is www.example.com? It finds it. It knows it. It puts it in memory. Now do I know the answer yet? Yes.

Now I know the answer. The thing that's most interesting about this is we estimate between 5 and 10,000 lookups go on before you need to look at the root server. So, all of us who took a computer science course learned that the first thing that happens is you look up something from a root server. And it turns out in actual practice, it's really closer to one out of five or ten thousand times.

So, this I thought was interesting, I hope you find it interesting, it's kind of the point of putting all the rest of this together. So, again, a zone is basically a file that contains a whole bunch of addresses and

names. So, there are 350 million domain names out there. It's a big number. There can be lots of sub-addresses to them. I recently recognized the case where you can be a company name dot com, and every one of your customers gets something to the left of that.

I don't want to name any of the people who do this, but you know it's mycompanyname.theircompanyname.com. That means they can have millions and millions of customers there to look up, and that all is within one of the 350 million zones out there that is maintained by the registrant. So, the registrant gets to determine where the addresses are, what's out there, and how many of these it has.

For the TLD zones, it's a much smaller number. There's 1,450 of them, and that is maintained by the TLD registry. There is one root zone, and it only contains about 1,450 files, and the root server operators have nothing to do with forming it. That is created and maintained by IANA and by the root zone maintainer. Does anybody keep track of time? I forgot to see. How bad am I?

CLAUDIIA RUIZ You have four minutes left.

JEFF OSBORN Oh, excellent. Thanks. So, in review, the root zone basically holds the addresses of the authoritative servers for the TLD. That's it. Just the TLD. We don't have any mail information, we don't know what website you went to, we don't know anything. This is the

equivalent of handing out a country code. If I tell you the country code is 44 and you come back and tell me you listen to what I said to my cousin.

It has nothing to do with it. We're literally giving an address to a top-level domain. So, we're really, really far removed from having any information, we're just handing out an address. The TLD is authoritative server knows the address for the next step up and the domain name authoritative server for the one above that. This is my t-shirt slide.

I'm going to buy the t-shirt and give it to Warren and hope he wears it. In the millisecond world of resolver, queries to the root server system are rare. And that's not intuitively obvious if you were awake during Computer Science 101. Root server operators do not decide what appears in the zone, registrants decide what appears in their own domain, TLD registries determine what is in the authoritative server address for them, IANA puts it all together, the root zone maintainer cryptographically signs it so it can't be changed, and all we do is deliver it.

So, in summary, we provide server information, not content. We don't decide what information is in the zone, and we're not a gatekeeper. I've had people tell me I have a latency problem because the root server isn't close enough to me. The odds of knowing that one of your 10,000 queries had R delay built into it, it's kind of difficult to imagine how that would ever happen.

The system itself is resilient, there's no single point of technological failure, we all use different operating systems, different hardware, different routes, different locations. And institutional failure is not possible as recently happened sadly. No force majeure can affect any three or four of us. That was an hour presentation in less than 20 minutes. Okay, good. Thank you. I appreciate your time.

LUIS DEIGO ESPINOZA: Thank you for the presentation. We have a couple of minutes for questions. Nigel.

NIGEL ROBERTS: Yeah, this one's for Steve, actually. Hi, Steve. Nigel Roberts for everybody else. I think Steve remembers who I am. I'm going to cast my mind back more than 30 years before RTLD existed, at least. I used to work for the phone company. And a lot of what you said actually reminded me a little bit about the considerations for important information in the databases of telephone subscribers, as we used to call them.

So, you'd start out, you'd have the phone book, and people would put stuff in the phone book because there was no reason why not. It was a thing. And that was your ordinary subscriber. Then you'd have what we called in the UK X directory, in the US the terminology would probably be unlisted or something similar.

So, the example for that is a doctor's home number. There were a couple of levels that perhaps weren't well aware to the world. We

had one called NQR, No Quoted Reference. The example for that might be royalty, the security service SIS CSG, Cheltenham.

And there were perhaps rumors that there were other levels of data, including phone lines that the phone company didn't know existed. And there seemed to be a bit of an echo here. If there's anything we can do to help you with this project, just give me a shout. We're happy to help.

STEVE CROCKER

I'll willingly accept it. Thank you. So, we'll talk offline, Nigel. Just very quickly, one could ask, so what's different between the phone books of yesteryear and DNS today? One is enormous automation. It was very much harder to take what in the U.S. we would call the white pages and generate spam. Now, it's not and things just progress from there.

LUIS DIEGO ESPINOZA

Well, thank you. Let's move to the next presentation. Jaromir Talir from CZ.NIC, DNS Hosting with FRED. Go ahead. Thank you.

JAROMIR TALIR

Hello, I'm Jaromir from CZNIC. I wanted to share via Zoom because I want to do a live demo at the end. So, I don't know how to get rid of these panels, but I hope you don't mind. So, what is it about? Many of you are registry operators, we are a TLD operator, which actually means to maintain the database of the domains and the

link to the five DNS resource records, so SOA, NS, DS, ANquad A, and to have those own file generator that generates its own file.

So, we have always thought about like why not to extend this possibility to all DNS resource record? Wouldn't it make sense to make the system more generic? For a long time we didn't have a opportunity to implement this, but we got opportunity last year, so this is the story about what we did when we start to manage .gov.cz domain, governmental domain and we moved the thread with the features more also towards the DNS hosting operators.

But I had couple presentations about FRED already. It's an open-source solution with all the features. There is a website for that. You can download the source code, you can download the packages, and definitely we also provide help and next business support if you wish. The system is used by the many countries. Our most recent addition is Venezuela that we actually find out just by accident or we are told at the last ICANN meeting that the people in Costa Rica helped people in Venezuela to establish the registry with FRED.

So, it's great that the community is growing and that those people can help themselves and each other to install and run system. The most recent changes that we published in the last public release couple weeks ago is that we are continuously moving from the binary packages deployment to deployment based on the docker containers. So, there are three more components that are

available via docker. For the remaining packages, we switched to Debian as the only platform that we now support.

We used to support Ubuntu and Fedora, but there was a lot of maintenance for the maintaining packages. We upgrade to more recent database engine. We also support the BSA global block module for couple registries that are using FRED. They want to connect to this platform, and there is also update that we have the virtual image that you can immediately download and deploy just for demo purposes to run FRED.

But back to our gov.cz project. Now in the period of time where it's quite common to do the fishing against many websites, governmental websites are not excluded from this. The Czech government decided to move all the websites for the government under the gov.cz domain. It was not the case for many years. This was the part of the unification.

They also decided to have the unified visual presentation for the websites, but also the part of this project was the transition to the gov.cz domain. Of course, they made a tender for provider of the registry for gov.cz domains and we applied. Why not, we know the business, so we applied, we won, and in the August 2024, we signed a contract with the Digital and Information Agency who is responsible for this digitization stuff, we call it DIA.

And we started to work on that and we delivered the final part according to contract in the December last year. So, what was the scope of the contract? It was two parts as usual to have the DNS

infrastructure for gov.cz. So, we moved the gov.cz to our Anycast. Already a year ago in December 2024, we transferred all the domains there to our systems. And for the year, we did a manual processing of DNS record changes via ticketing system before we were able to provide a second part of that which is like full registry with the main feature is the web portal for DNS administrators of these governmental institutions where they can easily administer their domains.

One part of this was that in the portal, we have to use the authentication portal from the government because the government has the central authentication system where all the employees are registered. The other part of the contract was that we need to have the API for automated DNS administration. Mainly what you can think about is like when they need to Due to ACME for the TLS certificate issuance, they need to API to put these Unicode into the DNS.

And we also have a request to solve this dynamic DNS changing, where some companies are using these global server load balancing systems. So, we needed to support also this. And we deployed the test instance in November 2025 for govtest.cz which is like complete copy of the instance for the production gov.cz which we run in December 2025. There is a mistake, so 2025, last year.

This is the schema of the system. On the right side is what we are running in our premises, and the left side is what is run on the

premises of DIA. This is the part on the far-right side, there is the DNS Anycast that we are operating for the domain. There is a part is FRED. You can see here that as FRED is quite modular system, we are not using all the parts of the system for GovCZ. For example, we are not at all using EPP module.

We developed new component for FRED, it's API that can be used for doing the changes in the database. And this API is directly used by the portal. The portal itself is not part of the FRED, it's specifically tied to the government use cases by the design and by the connection to this central authentication system. So, this is a high-level architecture. What's changed is the domain life cycle. I already mentioned there are no registrars of this system, there is no EPP.

DIA is the only single registrar. Its employees, DIA employees and the employees of the governmental institutions, they jointly administrate the domain registry through the portal. And the registration procedure starts with the request, that the requester must provide the domain name, some reason in the note, phone number. And then the organization which is represented is signed, logged in user, will be the owner of the domain, the holder. DNS configuration for all these domains is the same.

It points to our name servers. Internally we have the database of all the governmental institutions that we are syncing to have the database of all the contacts that can be the holders of the domain. The expiration date is set for 10 years. It doesn't matter, but we

have the -- this is the feature of the FRED that they can be maximum 10 years registered, but we implemented auto renew, so they don't actually have to care about the expiration date.

And this registration request must be approved or may be rejected by the appropriate administrator in DIA. And of course, one of the features is send the notifications from the system to all the parties to know whether the request was rejected or not.

There are two specific roles. One called DNS admin is the administrator in the governmental institution, and they are allowed to request registration and they are allowed to modify DNS record for registered domain and then something we call DNS super admin which is usually the employee of the DIA, and they are allowed to approve or deny request.

But they are also allowed to directly register some domains on behalf of themselves. The user that is signed into the portal, they always have the assigned organization and they can administrate only the domains that the organizations own in the registry. So, we had to build the arrest API for all the functionality. I mentioned this is part of FRED, so anybody who is using FRED can use it. it contains full functionality, it has access control by tokens that can be created and scoped.

There is a detailed documentation, you can go to the link that are in the slides where you can see all the documentation. And we implemented the support for this specific use case for ACME as a server plug-in that you can immediately use to connect to this our

API to is process of TLS certificate issuance. This is all the DNS resource record that are now possible to administer to the domain in FRED.

We are checking some syntaxes. We had to check like the collision if there is a CNAME on the GovCZ and the subdomain at the same time to avoid these kind of collisions. And for this dynamic DNS issues, we decided to implement ANAME virtual record which is inspired to the already expired draft where it's possible to create a name record. And there is a script that runs every minute that resolve this ANAME record to A or QuadA records. And it doesn't change anything if this resolving fails.

So, it should work for these cases where the organizations are using some systems like for example bit Azure Front Door, which is quite often used for governmental institutions. There was a small problem. First, we implemented that, we decided like to only resolve the DNSX signed domain.

However, Azure Front Door domains are not DNSX signed, and it's probably nothing we can change about that. So, we have to relax a little bit and allow these domains also to be part of this resolution. We need of course to do some changes with the zone file generator to handle a new DNS resource record type, and we also implemented possibility to generate the catalog zones for the all configured domains.

So, FRED now is able to generate the catalog zones for the domains in the configuration. For here, these third level domains under

GovCZ each have to separate in zone file so it matches the requirements. For the GovCZ we run every five-minute propagation of changes. It's quite easy, like there's right now about around 400 domains.

And we maintain the DNSSEC automation normally because we run no DNS on our Anycast or on the master DNS server, and we have the FRED which is also supporting that, so nobody needs to care at all about DNSSEC in this system. Of course, we provide the directory services which is something which is already implemented in FRED like Web WHOIS, RDAP, so you can query who is the owner of the third level domain easily. What we plan as the next steps?

There are some few updates that we are discussing with the SDA, better scoping of API access tokens because they are allowing the other parties like the service providers to access these DNS operations on behalf of them, so they need to scope them on the level of domain. We scoped that before on the level of organization.

They want to have some more hierarchy in these contacts owners, because they are mostly ministries owning the domains, but they have some subordinate organizations. They want to show more history in the portal for DNS record changes, and we are also talking about the possibility of redirection from the .CZ RDAP to GovCZ RDAP, which is not yet implemented.

I already mentioned that portal is not part of FRED, so there are some features missing right now. We believe that the functionality of the portal should be integrated into our own internal administration or we are thinking about maybe built another generic portal for other customers on the same API. All these API and the backend for this DNS hosting will be part of the next public first release.

So, we will announce it if you follow the website, we will announce it so we can test it yourself. We are thinking about additional use cases, what we can do with this. Of course, we actually already have some VIP domain commercial service that we provide directly to the customers for the high-profile domains. We have T-Mobile, Czech Television that they decided to host domain on our Anycast.

And right now, we are only like the secondary hosting. They provide wire zone transfer. They provide us the zone file. So, if you provide them self-service portal, they can immediately edit the zone directly in the system. And of course, we have our own SLDs, about 100 of SLDs for our own project. So, right now, we administer all these zones in the gate and changes are done via ticketing system that our system administrators are doing changes directly.

So, this could also change if you would provide the possibility to have it directly in FRED. But for now, like, so this was about this project, but right now our priorities are more focused on making FRED compliant with the ICANN requirements for the new gTLD. So, we are participating in this ICANN RST program. We created a

new RD exporter to be aligned. We did almost finished RDAP alignment and even EPP alignments through the proxy. So, the goal should be that FRED could be easily used also for the new gTLD.

And running this GovCZ registry also gave us quite valuable experience in hosting FRED environment because right now we have not provided any hosting FRED version. So, that was about the portal, and I try to switch to just simple demo. This is how the portal looks like. So, I should be able to sign up. Everything is in check, but I think you will understand. This is the CAS, the Central Authentication System that DIA is using.

It's actually redirecting to our government's digital identity solution. Ah, they have outage. That's the government. So, let's try it once again. Hmm, it's worse. So, for authentication, I will use actually our own digital identity because it's part of the governmental structure. So, we are running the digital identity as well and the portal at the beginning. So, we have covered from both sides.

And here, this is the authentication. No. I'm still on the website of the government. So, probably a lot of people has a problem in Czechia right now that the main authentication system is not working. So, once again. So, if this will not work, then probably -- okay, so we didn't get to the portal yet.

So, there is a complex network of the authentication features that needs to be needs to get through for the authentication to get

through. So, it's a pity I cannot show you the portal because I cannot sign into the portal. Just last attempt and I'm stopping. Okay. So, I'm not showing anything. Never mind. Thank you for watching. Sorry for our government.

LUIS DIEGO ESPINOZA: Thank you, Jaromir. Let's give an applause to all the presenters. And we can take questions. Yes, over there.

MAARTEN AERTSEN: So, this is Maarten Aertsen with RDAP for the record. So, let me save you from the demo gods and ask you a question. So, the Dutch government is currently tendering for something you've built, and it greatly interests me that you have built this on top of FRED.

So, I was wondering if you were aware of the fact that they are tendering what you have built and whether you are looking for more European government infrastructure to maybe run, or would you be interested in talking to these people? Like obviously they're tendering so, yeah, it's like a process but still.

JAROMIR TALIR: Yeah, we are interested evidently to hear more. I suppose that most government will have specific requirements probably that could not be so easily to tidy, but let's talk about it. Yeah, we are interested.

LUIS DIEGO ESPINOZA: I have a couple more questions. This one first. He has first, yeah. Then next to you, and we close. Thank you. Go ahead.

PETER THOMASSEN: Hello, Peter Thomassen. So, I think this is pretty cool, so thank you for presenting that. You said that for gov.cz to use CDNS key automation for the third level registrations, and I think initially you showed there are maybe 13 ccTLDs that use FRED, and certainly all of them support DNSSEC. But I am only aware of two that actually use the automation, which is I think .CR and your TLD. So, I was wondering what's the default setting in FRED for the DS automation, and have you considered turning it on by default?

JAROMIR TALIR: Yeah, it's not that easy as usual. The DNSSEC automation is a specific part like as a module. So, this is like module that you have to install and configure. So, if you don't want DNSSEC automation, you don't install the module. So, it's like nothing that we can actually from our site to put as a default. So, definitely we are trying to talk to these people and encourage them to install this module and to run it, but so far, as you say, only Costa Rica was active and implemented that. So, it's about little bit additional work than just to switch on.

LUIS DIEGO ESPINOZA: Calvin, last question please. Thank you.

CALVIN BROWNE Yeah, Calvin Browne. Sorry, excuse me, I'm balancing my Red Bull intake with my jet lag. I just want to check, so that portal is almost like a WHMCS type thing. Is that kind of how I could read it but very custom for the Czech government project?

JAROMIR TALIR You can think about it, but without all these, like the billing thing, like it's editor for the DNS resource record let's say with all the functionalities like requesting changes and approving changes.

CALVIN BROWNE I wasn't quite sure if you had a plan to release it separately or…?

JAROMIR TALIR Well, the portal is just a simple layer on top of the REST API. And the portal is mostly tied to the design of the government and the authentication from the government. So, as I mentioned, we plan to maybe have the generic version of this portal, build a generic version of this portal, or integrate this portal into our own web administration. So, we are thinking about what would be better. But actually, you can do, I think that this UI generated from REST API could do it the same way.

LUIS DIEGO ESPINOZA	Thank you very much. We are closing this session now, and be back in half an hour. Thank you.

[END OF TRANSCRIPTION]