MONTREAL – How it Works: Understanding RDAP
Sunday, November 3, 2019 – 10:30 to 12:00 EDT
ICANN66 | Montréal, Canada

STEVE CONTE: Alright, we're going to go ahead and get started, you can start the recording, we are, good. Thank you guys for staying. This session is going to be on RDAP, understanding it, and knowing where we're going with it. We have Francisco Arias from the Global Domains Division of ICANN and I'm going to pass it over to him. We're going to take questions as they come up. So if you have any questions, raise your hand, we'll get a mic over to you. And then we'll probably have some time at the end, as well, for additional questions. So, without further ado, Francisco.

FRANCISCO ARIAS: Thank you, Steve. Hello everyone. This is a one-on-one session on RDAP, so this is the protocol [inaudible] to replace WHOIS. Let's go directly into this. This is the agenda for today. The Introduction, then comparison of WHOIS versus RDAP, the output. I want to explain what the RDAP Protocol is, then talk a little bit about the gTLD RDAP Profile which is important in the context of gTLDs. Then Next Steps, and then at the end, I prepared a demo for you to see what RDAP actually looks like. As Steve said, please feel free to interrupt me if you have any questions. My colleague Gustavo here at the front has a microphone and he can walk around with it, so you can ask. So, let's start.

First let me clarify something. There are at ICANN two lines of work that are going in parallel, so it may be confusing, so I want to clarify what this presentation is about. There are two things going on in ICANN. One is the retirement of the WHOIS protocol and the introduction of the RDAP protocol.

So the first line of work is the one about introducing RDAP and retiring WHOIS. This is something that has been going on for quite a while in ICANN, the work started back in 2010, I want to say. This is related to user needs and limitations on the WHOIS protocol, a lack of standardization in how you query, how you get responses or messages, lack of internalization, etc. etc. So this started a long time ago.

The other thing that is going on in ICANN is policy regarding the processing of registration data, like the EPDP, the Expedited Policy Development Process that is going on, you may have heard of Phase 2 of that policy, the implementation of Phase 1 is already ongoing, I have a little bit more on that. At the moment there is also the temp spec that was passed a couple years ago. So, all of that work is related to the information that goes inside the protocol, be it WHOIS or RDAP or anything else.

Now, I must say this presentation is about the first one, just about number one. So if you were expecting to hear something about what the policy is, who will get access to what data, what data will be shown and all of that, I'm sorry to disappoint you, but this is not about

that. This is a little bit more on the technical side of what the new protocol that was just introduced is all about.

So, this is a simplified diagram of the Registration Data Directory Services, or you could say at a high level of the domain name environment. So we have the registrant, who is the party or person that is registering a domain name. They talk to a registrar, the registrar talks to the registry and the domain name gets created. And then we have until before August we only had WHOIS as a way to access information to see who had raised there a certain domain name to see the data about the registrant, the contacts, et cetera. And by the way, we also have the RARs, they are the ones that deal with IP addresses, autonomous system numbers. So the other side of the identifiers in the internet. They also have been offering WHOIS for quite a while.

And so what we're doing now is introducing RDAP as another a protocol, another option to obtain the information. And the idea is that eventually the WHOIS protocol will be transitioned, will be phased out, and it will not be available anymore. But at the moment, the situation we have since August is that both protocols are available, so you can access the information, the same information in two different ways.

And as you can see here, there is no ICANN in the picture. ICANN is not involved in the passing of the information. The information is directly accessed from the registry, the registrar or the RIR. ICANN is sitting somewhere there dealing with policy and compliance and other stuff

that we do, but ICANN is not involved, at least not at the moment, in the passing of information, accessing the information from the registry, registrar, or others.

So, as I said, the work for replacing WHOIS started long ago, it has been going on for quite a while. There was work in the ITF, the protocol was finally standardized, the RDAP protocol was finally standardized in 2015 and since then the RIRs, there are five of them, they started using RDAP and they have been using it for years now. They are the leaders in this field. Since then, some ccTLDs also started offering RDAP, I believe there are a dozen or so ccTLDs that are offering RDAP already, and since August this year, so just a couple months ago, all registries, gTLD registries and registrars are required to offer an RDAP service and most of them are already doing it.

There is still more work that is needed, for example there is a requirement to have a common profile for how the RDAP service is offered in the gTLD space. At the moment if you don't have that, then you will have a slight difference in the output in the gTLD registrars and registries, and you probably don't want that. We want to make the life easier of the users so they can know what they can expect when they are querying an RDAP server. We also need to have explicit, production-quality RDAP SLAs. There is some work that needs to happen in regards to ICANN negotiating with the current bodies, the registries and registrars. In that regard, I believe there is a closed session on the agenda for those negotiations.

There is also the need to have some reporting requirements regarding RDAP. This is similar to what already happens with WHOIS, EPP and DNS, registries report to ICANN on a periodic basis. The number of queries they receive, so this is something that helps ICANN and others, because this information is eventually made public, understand what is going on in this space.

And of course the last step will be to eventually retire, the WHOIS service, the 443 service, since it has so many drawbacks. ICANN has an RDAP page where you can find some information about the protocol and how to use it if you're a user, and there are also some resources available for implementers, that's, the registries and the registrars.

So, a brief summary of the features that the RDAP protocol offers are basically a solution to all the drawbacks that WHOIS had. So, you have a standardized form of query, which is, by the way, HTTP, because RDAP is protocol that runs on top of HTTP. HTTP for those of you that don't know is the protocol that is used in the web. It is the most used protocol in the internet and is something that almost any company in the world will know how to do, or they will have no problem finding someone that can run a web page for them.

So, that's the way you do queries. There are of course some nuances to that, but that in a sense is what is there. The response is returned in JSON. JSON is a way to provide plain text responses with weight encoded, we'll talk more about that later. And error messages are also on the rise using a combination of HTTP error messages and more JSON where you can more information, there is more about that later.

With RDAP you can also secure access to the data. You can have the servers use HTTPS which is the standard use in the internet, in the web, to secure access to the data so the client can know they are talking to the party they intend to talk to. And there is also confidentiality in that communication. The protocol is extensible, you can add query, you can add support for other objects. RDAP is quite extensible, and was designed with that in mind. It enables differentiated access, so for example, having the ability to limit who gets to see what information. Of particular interest is parallel that it ongoing in ICANN, the EPDP, for example the EPDP Phase 2 is all about enabling differentiated access.

So for example you could limit access for anonymous users so they have only the limited set of fields that are described in the temp spec and now in the EPDP Phase 1 policy, and offer full access or some form of access that is beyond the limited set for authenticated and authorized users. So this is something that RDAP enables you to do. And by the way, RDAP is like a menu, it tells you how to do things, but it doesn't tell you which features you have to turn on or not, that is a decision that is left for the policy, for the implements.

RDAP also offers what is called a bootstrapping mechanism, a way to find the authoritative source of the information. In WHOIS you need to know what website, if you're using web whiz or what server in Port 43 to query in order get the information. In RDAP, you do not need to know that. If you have an RDAP line, you just tell the RDAP line I need to know information about this domain name, this IP address, this

autonomous system number, and the client takes care of finding the server and then presenting the information to you.

RDAP also offers a standardized way to do redirection or reference mechanics. So for example, in the space where we have registries and registrars, so the registry doesn't have all the information about registration. With RDAP you can have reference mechanics so the registry could say here is the information I have about this domain name and here is a link to the RDAP service of the registrar for this domain name where you can get the rest of the information that the registrar has. So, you can do that in RDAP.

As I said before, it's built on top of HTTP, which makes it quite easy to implement. It has internationalization support from the start, it's a no-brainer there. It has support for Unicode, the standard coding in the internet. It also enables to do searches on objects. You can find for example the domain name by someone, or that will start with a certain letter, something like that. Of course this is a very power feature and is subject to policy to define if that should be enabled or not. I'll I'm doing here is describing the features, which ones are turned on or off depends on policy decisions.

So, a quick description of where things are on the policy development, just so that you have an idea where we stand there. As I said, there is a phase one of the EPDP which dealt with the processing of data, finding how to do the collection, the transfer between registries and the registrars, the publication done in WHOIS or RDAP, et cetera, and the

implementation of that one is still ongoing. There are some meetings happening here in Montreal.

There is a phase two which is still in the policy development phase, that's where they are considering enabling some form of differentiated access for the registration data. I understand they intend to publish the initial report by December, although I heard in a session yesterday that may be delayed a little bit, but I'm certainly not the source of that information. And so there are a few sessions may be of interest to you if you're interested in policy.

On the policy side there is a plenary session tomorrow. You can find information there. And there are four EPDP Phase 2 sessions that are working sessions where they discuss the future policy. There are also two EPDP Phase 1 implementation sessions. Those are also working sessions, because the policy has not been finalized and so you may find that also interesting. There were so many sessions, I could not fit them in one page. But if you go to the schedule, you won't have difficulty finding the different sessions in case you're interested. The last bit on that that is not exactly policy, it's actually not policy at all, but it's related to the policy because it was intended to inform the discussion in the policy development process.

There was something that ICANN Org did which is launch what is called a technical study group that dive into the technical solutions to have differentiated access service in RDAP. So there was three or four months of work with technical experts that were invited by our CEO from ICANN and they published the final report last April that

describes a technical model that could be used for what it's called a Unified Access Model. So differentiated access in RDAP. Of course the policy and where that is going to be implemented is something subject to the discussions in the EPDP Phase 2. This is all I have regarding policy and I don't have anything more. So I'll just come back to the RDAP stuff.

So, differences between WHOIS and RDAP responses. One thing they have in common is WHOIS and RDAP are both text based, they return plain text responses. One difference is WHOIS does not define too much about the query of the response, it's just bytes are returned, period. That's what WHOIS does. In the case of RDAP, the response is highly structured and is focused on being machine parsable. So I'm going to show you some examples later and you're going to find that fortunately RDAP is not intended to be human friendly.

Now having said that, it's because it's machine parsable, it's quite easy to be converted to something that is human friendly and some people, including us, have done just that. We have a web page where people can just go to that, look up ICANN Org and you can use RDAP without even knowing. So, RDAP is also flexible in terms of the fields and the functionality that can be added, I guess I kind of covered that before.

So here are some examples. This is the WHOIS output. This is what you get, the examples I'm going to show here, they are for the exact same domain name, icann.com, so it's a query to the VeriSign WHOIS server. All there is there, all the formatting, spaces, et cetera, that's

how things have to be in WHOIS. You need to add everything in order to make it look the way you want it. With RDAP, it looks like that. Not exactly human friendly, as you can see, but this is very easy to parse for a machine so it can easily convert it to something like this.

Or even more usable, something like this. This is the actual output from our web client. This is the lookup for ICANN.org webpage if you go there, as I said before, you will be using RDAP without even knowing it, and there you can see the information in a human friendly way, even has at the end the option to see the roll out of the output in case you were interested in seeing that. We'll talk more on that later, in more detail.

So, the protocol, let's talk more about the RDAP protocol what that is. The RDAP protocol as I said before was defined in the IETF. The IETF is the organization where most of the standards using the internet are defined and maintained. They have a series of documents called RFCs and the RFCs are where the standards are defined. In the case of RDAP, these are the relevant RFCs that are available at the moment from the IETF that describes them, they are the protocol.

For example, the one you can see at the bottom is already an extension we had to define in order to enable a couple features that we needed that are unique to the ICANN ecosystem, and it was quite easy to add that extension in the case of RDAP. So, a little bit more about the IETF, well, I guess we don't need to cover too much detail, but that's the organization that takes care of most of the standards in the internet, the technical standards, that is.

In the IETF there are working groups that deal with certain topics. In the case of RDAP happens to be the same working group, the so called REGEXT working group is the one that deals with standardizing EPP, RDAP and their registries, registrars and RIRs participate to do this work. Now, getting back to the protocol, there are two main concepts that you may be interested in understood here.

The first one is Lookup. Lookup is what we usually do in WHOIS and what we usually do in RDAP. Lookup is a query for a specific object. So when you're looking for information about a domain name or an IP address, you are doing a lookup. You give the RDAP client or the WHOIS client the domain name you're looking for, so the protocol will take care of finding the information about that specific object. That's Lookup.

There is another type of query which is the search, which I touched briefly before in the presentation. It's a query where you want to find objects, for example domain names, that have a certain common characteristic, that were registered by a certain organization, or registrant, or contains a word, or have a name server, or something else. So, the search, as I mentioned before, is a very powerful tool, but also something that could be very taxing on the server, so they have to be used with care, let's say, and of course the searches are subject to policies to dictate where they should be enabled or not, and what is the extent of the searches, it's quite complicated to get it right, this thing.

So, in terms of the Lookup, which is the main thing we will be concerning this presentation, in RDAP there are five objects that are defined in the base protocol. Of these five, there are two that are probably the most important, at least in the context of ICANN. The first one is the domain names and we probably would be interested in finding information about a certain domain name. So you can certainly do that with RDAP.

IP addresses or networks for the matter, they can also be searched in RDAP, but there are support for other objects, name servers typically linked to a domain name, the autonomous system numbers, that's like an identifier for a network. And entity, this is an abstract concept that was put in RDAP to encompass people and organizations. So an entity can be a person or an organization. So the contacts, registrants, registrars, resellers, they all are mapped to an entity in RDAP, that's the name that was given in the protocol.

So domain queries, this is where you construct a domain name query in RDAP. You have a base URL that is dependent on the RDAP server you are querying. But remember, this is something you probably will not need to do, I'm just presenting this in case you're interested in knowing, but you're using an RDAP client, you will not need to know all this stuff. You don't need to know how to build the query by hand, you don’t need to know how this works.

But in case you're interested, this is how it works. You could try this as a matter of fact in a web browser and it will work, it will give you the RDAP information, because it's HTTP. But if you have an RDAP client,

you don't need to worry about all those details. You just tell the client, give me information about this domain name, and that's what you will get. Now the domain name can be In ASCII form or Unicode if it's an IDN. The protocol of course, it has support for all that from the start, there is no issue there.

The responses, as I said, they are encoded using JSON, JSON means JavaScript Object Notation. So another standard from IETF, the objects in JSON are a pair of name value pairs, so you have the name of the field and the value of the field. In that sense, not very different from what WHOIS does, it's just that you have an extra set of things like the brackets and you have the columns separating the values.

The interesting thing is in JSON you can have more structured data, you can have types in the data. So you can have numbers, strings, Boolean, so true or false data. You can even have arrays, which is a list of things, for example a list of strings or a list of numbers, a list interested in the case of domain names, for example, to have the statuses of our domain name because a domain name can have more than one status. So you have an array, and that's what you store all day. You can have an object nested inside another object, or it could be known.

Just an example of JSON, just so you see how it looks. We are not going to get into details there. And in going back to the domain name response, you have these members, you have these elements that are returned in the domain name response. So you have the handle which is the domain ID, that's typically what researchers use to ask the

handle for the object. IdhName is the name of the field for the domain name in ASCII. Unicode name is the name for, if it's an IDN, that's where you will find the Chinese characters, Arabic, Latin with accents, et cetera.

Of particular interest, perhaps, those are the name servers of the domain name and entities, as you can see that's an array also, and that's where you will have all the contacts of the domain name, so you will find the administrative contact, registrant, registrar, reseller if there is one. So all of those will be there in the domain name response. You can find them in there. Finally, the last set of data is the data related to the DNSSEC, I'm not going to explain all the fields, but this is all the data related to a domain name that is DNSSEC signed.

So, in the domain name response, I said you get the context registrar et cetera, related to the domain name. So we need to know what is the structure of the entity in order to know that, because the entities will be nested in the domain name. So, for the entities, you will also have a handle so the contact ID or registrar ID, or whatever the entity is. You will have a set of roles. A role indicates what rule the entity is playing related to a given object, the parent object.

So, for example in the case of a domain name, you can ask that of entities, and how do you know which is which, which is the administrative contact, which is the technical registry, et cetera, the way to know that is with the role. So you will find a role for the registrant which will be registrant, so you know that entity is the

registrant contact. You will find another role which will be registrar, and so you know that's the registrar data, and so on and so forth. I think those are the most important fields in relation to the entity. Oh, no, I was forgetting something pretty important, number two there, vcardArray, it doesn't seem like much, but that's where you find the actual contact information.

In the EITF there is a standard for representing information related to a person or an organization, it's called vcard. Vcard is an old standard that has been around for quite a while, you use Outlook or pretty much any other email client or contact, so contacts, they are using vcard most likely. It's the standard to manage that. So IETF created a mapping in JSON which is called jCard, and that's how the information of the person's organization is stored. So you have an array there, so for example where it says fn that means full name, that's the way vcard works. And so you have to live with that. So fn means the full name of the person in this case or organization tell the telephone number. And then you have the address.

In the case of the address, you don't have an indication of what is each thing, but there is an order, the order you can see it on the left side, so you have the post code, the address can be the field number to extend an address, that's one possible way in which you can have the address which you just have a block of text there that will include for example the number the street, the city, the country, all of that. In the utility space we are pushing to have the structured way to have the addressed, which is having the number field empty and the rest filled in, which is more like we have now in WHOIS, which is where you have

the street address in one field, the city in another, the region postal code and country name, etc.

By the way, the country name is one of the things that remember I mentioned before that we had to create an RDAP extension. The RDAP extension that was created was in relation to using country codes, instead of country names. And so one of the things that we want to have in a level one gTLD registries and the registry users, they use country codes as is being done with WHOIS, instead of using country name, which brings a set of issues that are outside the scope of this talk. So, that was the domain names and inside the domain names the entities that I already talked about.

Name servers are very similar. That's how you create the URL for the data. Remember, if you have an RDAP client you don't need to know this, but in case you wanted to know, this is how you do it. And again, it has support for the ASCII version or the Unicode version, in case of an IDN. And in the response you get again all the objects have to have a handle and LDH name like in the domain name cases, the ASCII name, Unicode name, same thing. You have the IP addresses and you can have entities linked to the name server. In the case of IP addresses this is how you query.

As you can see there, there are two notations that you can use. You can use an IP address by itself. You can add a block of IP addresses and you can do IPv4, IPv6, all of that is supported, of course, and there are some examples below on how you will do the query. What you get in the IP address response, again there's a handle. There is a start and

endAddress, the reason for that is because as I said before, you can query lots of IP addresses or you can get a response that ends up being a block of IP addresses. So you have to know the start and the endAddress of the block. The version before v6 and a few other things related to the IP address.

Autonomous system numbers, this is just a number, I believe it's now 64 bits, what, 52, okay, thanks. So, it's a number that it's used to identify the identity of a network. And like IP addresses, this can be a block of autonomous system numbers. For example, when an RIR delegates a certain block to another entity, for example, sonar RIRs, they have in the registry, a specific country for example has a registry that is in charge of delegating IP addresses to the ISPs within that country, so they could be delegating a whole block of autonomous system numbers. So that's for the objects that are returned in RDAP.

The other important thing to mention about RDAP is their error responses. So these are codified again to be easy to handle by a machine, so they are very structured. There are two things that you get in a response you get the HTTP response code, they are highly standardized and currently used in the internet, of course. So they indicate, for example, an HTTP error 404, it's what in the internet indicates that certain resource, for example webpage, is not available. So part of that if you get a 404, it's indicating that the domain name is not registered. But no not only get the HTTP error code, you also can get JSON response like the one shown there as an example that gives you more information in case the server would like to explain more

about the error and situation that is happening. That's for error responses.

So, as I mentioned before, RDAP protocol gives you from start your need to enable anything, it gives you internationalization from start. It was decided to be internationalized from the beginning. As I said, it supports Unicode and more specifically, UTF-8 encoding, and it supports internationalization in two senses, internationalized domain names in the query and the response and also the contact data can be internationalized.

So if my language were to be something different to English and my name or my address, city, country, whatever, were to be using something that is beyond ASCII, so it could be for example here in Montreal, you will use perhaps accents. You are using Latin characters, but you may be using accents. So that is supported in RDAP. Of course even other scripts like Chinese or Arabic, et cetera, all of that is supported in RDAP.

The contact information could also add language tags in order to identify the language or the script of a certain field and this may be something coming to I believe translation and transliteration policy. That is something that is still to be implemented, but it has some recommendation that needs to be implemented in regards to if the information is translator or transliterated that there has to be some language tag to define the language or script that was transformed. So for example, if the language was Chinese and was transformed to

Latin script, so it will have a language tag that will indicate that that happened.

What else is here, I think I already covered that. As I mentioned before, you have bootstrapping, you don't need to know if you have RDAP client, you don't need to know the specifics on how to build the queries, because all that works by the protocol itself. So, in order for this to work, IANA maintains lists of base URLs and so there you can find the base URLs for the RDAP service of all that TLDs, gTLDs or ccTLDs, and also for the IP address registries the RIRs, and the autonomous system numbers.

As I mentioned before, RDAP is highly extensible but also if we want clients to be able to understand what is being returned, they need to have an understanding of the RDAP extensions that are developed. With that in mind, there is an IANA registry that keeps track of the RDAP extensions and so people can know what is available that will be implemented by someone. Perhaps of interest here is this is the list of the some of the RDAP extensions that are going to be discussed in the IETF in the working group.

As you can see three of those, the first three are related to search capabilities and RDAP the base protocol as defined had very basic search, and some people thought they wanted a richer set of functionality in regard to search. And so they have been working on defining that functionality. That's still a work in progress in the IETF.

The last one, federated authentication that's a draft that is intended to offer functionality for making life easier for the users in the environment like the one in the gTLDs, where you have three thousand something registries and registrars and if in a future time when and if there is some form of standardized system to access the nonpublic information in RDAP, the users, if things are not changed the way they will work now, the users will have to go with each registry in order to obtain credentials.

So imagine going to three thousand something registries and registrars, that's probably a painful process. So this draft has the idea of having only one place where a user will go to register as a user and obtain a set of credentials, password, digital certificate, whatever, and use that to talk to any of the participants in that federated authentication system.

For example, the gTLDs, so that way a user, if this system were to be put in place, will allow a user to just go to one place and then be able to use the same set of credentials when talking with all the gTLDs registries and registrars. Of course, this is only the technical part of the solution. There needs to be policy that says yes, this is something that should be done, or not. So that's all I have in regards to the RDAP protocol.

Now, let's talk about the RDAP profile. As I said before, the RDAP protocol is like a menu, it says you can do this thing you can do this other thing, but it doesn't tell you what thing to do. That's something that is of course left for the policy to decide. So in the utility space

there has to be some policy or contract requirement that says this is what you do in RDAP. So that's what the gTLD RDAP profile is, it does the mapping of the current policy requirements to the RDAP implementation. So there is currently already an RDAP profile, this is a version that was developed in discussion group of registries and registrars, where ICANN also participated. It consists of two documents that are described there.

And as I said, this is still a work in progress in the sense that, at the moment, the compliance with the profile. It's our recommendation, but we don't yet have a way to require compliance with that profile. It is something that is ongoing, that are negotiations that are happening, even here in Montreal, so hopefully in the near future we are going to have a requirement for gTLD registries and registrars to comply with this common profile so that we can have a common output from gTLD registries and registrars. Now I should say that even though it is not a requirement, many of the gTLD registries and registrars appear to be already implementing that profile that was developed by this group.

So, what is prescribed by that profile? Just to give you an idea what are the things that are there. So it says what fields have to be included in the response. Also, it makes the differentiations according to policy and contractual requirements between what a registry and a registrar has to return. Registries and registrars, leaving aside the thin and thick discussion, even without considering that, they have slight differences in what is returned in their responses.

For example, there is a registrar expiration date that is an option for the registrars to offer the reseller. The status says the registrar may not know all the status because the registrar is the one that maintains it, so there are slight differences in the output of registries and registrars and the profile takes care of clarifying that for registries and registrars. The profile also defines what objects, so the queries that need to be supported, for example, registrars are required to support domain name queries, name servers and registers, and in the case of case of registrars, they only are required to offer queries for domain names. There's something that comes from their agreements.

Other type of requirements that are in the profile, they need to support IPv4 and IPv6 transports. There are certain HTTP headers need to be returned from the registry and registrar in order to enable things like what we are doing with our web client. If those headers are not returned then the web clients unfortunately cannot work. So it is something that is that profile that was published in February. The requirements for DNSX so that domain names that are used in RDAP are properly signed so that they can be securely confirmed that they are from who they are supposed to be.

What else is there, it requires registrars to provide their base RDAP URLs to ICANN so ICANN serves like a sort of clearing house that allows the registries to know those RDAP registrar based URLs to enable them to offer that. It indicates how to do truncation, truncation is a term that is used in order to indicate when our response has been relaxed. So since the temp spec back in 2017, there are requirements

for gTLD registrars to do with redaction of responses. So in the profile it says how you go about doing that redaction.

As I mentioned before, the use of country code instead of country name, that's something that is in the profile. There is a requirement in temp spec to have URL for a web form or an email to anonymize communication to the registrant, so that's another thing that's described in the profile, you do that in RDAP. What else? There is a requirement for example that registries only respond to the domain names they sponsor, so names that they register. This is in order to disable loops, because in RDAP given the function of reference and redirection, it could be problematic. The profile also describes that a registry provides redacted responses on the basis described in the temp spec.

I think we already covered the references. HTTPS, I think we accept this is the standard how the internet secures the protocol, you normally see that in a web browser with the green lock or some other cue from your browser that tells you that you're talking with a secure site. So that's something you get in RDAP although you probably won't have a lot there because I guess that's a client dependent feature.

I talked before about the redaction requirements. Reduction in other words a little bit different than how it would work in WHOIS. In WHOIS, redaction per the temp spec says that you keep the fields, even if they are redacted and you add certain texts to indicate it has been redacted. In the case of RDAP, you don’t do that. You remove

the fields, that's how the protocol works, if you don't have data, you don't have the field. But you add another element called a remark in the object.

So, for example, you're redacting certain contact registrant, you remove the fields that you're supposed to redact according to the temp spec. but you add a remark that says this contact has been redacted and you will need to have special authorization to see this information. And so that's how we will work in RDAP. And as I said, this is described in the utility RDAP profile.

Lastly, differentiated access. This is something that RDAP enables the term spec, just say that you shall offer this and describes the reasoning which are registry or registrar has to do this, but it doesn't describe a uniform mode to do this. As I mentioned before, there is future work that needs to happen in order to have such a thing, and this is something that is the subject of policy discussions more completely the EPDP Phase 2.

Finally, in regards to RDAP adoption, just to reiterate what I said, there is still work that needs to happen to adopt an SLA, reporting requirements, adherence to the common gTLD RDAP profile and eventually the retirement of the WHOIS protocol.

Okay, so this is the end of the slides. The only thing I have next is the demo. This is a list of the client implementations that we are aware of and you can find these slides in the in the ICANN website in the session webpage, so you can see the links to the different implementations

that we are aware of. As I said before, we have one implementation, which is a web client, so let me do a short demo on the these two tools, the website and the command line tool.

STEVE CONTE: While he's getting ready for that, are there any questions that I can jump to, while Francisco is preparing? I see no hands. Okay, thank you.

FRANCISCO ARIAS: Just to show you an example of how things will look in RDAP if you were to use our web browser. I'm accessing here the bootstrap registry for TLDs that is maintained by IANA, you can pick up this sort of like the root zone we have in the DNS. this is like the root zone for the RDAP servers in the TLDs. So I'm copying the base URL and I can do this, it's domain then domain name, let's see if this works. So, you can see this is the RDAP response. Not very human friendly.

The Mozilla Firefox browser has functionality that you can use to do sort of a more friendly output so you can see here the different objects. You can find, for example, the domain name the status, in this case only one status, when it was raised there, when it expires, when it was updated.

What else is here of interest? The data for DNSSEC, they have the DS record there. What else can we see here that is of interest? It says that it was redacted. For example, here is a contact that is a billing

contact. Anyway, this is how it looks, the RDAP output. Another way you could access the information using command line client. So with WHOIS with Port 43 the way you will do it is like that, right? You will use WHOIS and do the domain inquiry.

With RDAP there is at least one command line client that I know about, Nicinfo. This is something that was implemented by ARIN, so if you search your favorite search engine ARIN, nicinfo, you will find information how install this. As you can see, I'm not telling the RDAP client where to find information, it just knows where to find information, it's showing me for example the status, when it was raised there, when it expires, the DNSSEC information, who is the registrar, data about the registrar, the registrant which is not much, because it's redacted, there isn't much information, this record is redacted, name servers, et cetera. So, you can see all the information in a sort of human friendly way, but of course, this is a command line tool.

Now I can use a web client. So this is the web client that ICANN developed and this is how you would see the information. So, here you can see the information about the domain name and the handle, name servers, expiration, contact information. There isn't much about the contact information, of course, because it's redacted. Here you can see that it's telling you that the information was redacted. Information about the registrar, the NSSEC information, as I said before, and here is the reference to the registrar data.

Although I should say in the case of our web client, it does both a query to the registry and to the registrar and it matched the output to show it. What it does is the information about the contacts, so the registrar, the admin contact, that information is taken from the registrar response, and that's what is shown. The rest of the information, so the domain name status, the domain name itself, the name servers, DNSSEC information, expiration date, creation date, et cetera, that data is taken from the registry response. And so they are matched in our client.

But here you can find at the bottom, you can see the RDAP response from the registrar in this case, and this is JSON, this is the actual response that the registrar provided, and here is the registry response, so you could take a look at this if you wanted and you can find information returned by the registry and the registrar as provided by the RDAP servers.

So another interesting thing of the ICANN implementation on this web client is that ICANN, the way we develop this is using JavaScript. For those of you that don't know, JavaScript executes in your web browser. So what that means is the RDAP queries are being done from this laptop, not from an ICANN server. In the case of WHOIS, for example, this is a second generation of this tool; the first generation of the tool, the WHOIS, that was doing WHOIS queries.

For WHOIS, could not do it on JavaScript, so the way we work is some ICANN server in the background will be doing the querying, so we, ICANN, our servers will need to know which domain name you were

querying, do the querying in the background and then providing the output to you. So ICANN will be seeing what you're querying and the response you're getting. In the case of RDAP, because it's the web, we developed this using JavaScript and as I said that runs in the browser.

So, it's my laptop or this laptop in this case, that is doing the query, ICANN doesn't know what domain names you're querying, doesn't see the response, so we don't know anything about the query that is being done. So this is a nice feature that you get in RDAP in terms of privacy that can be enabled, that you cannot do in WHOIS. So, I think that's all I had in terms of the demo. So, questions?

STEVE CONTE: Thank you, Francisco. Any questions for Francisco on RDAP? So you're all RDAP experts now. Okay, question, good.

UNKNOWN SPEAKER: Hi, this is Mohit from India. My question is why was JSON used instead of XML?

FRANCISCO ARIAS: Good question. When the development of the RDAP protocol started in the ITF, that was as a matter of fact, the original intention, to use XML. The RDAP was based on something that was originally developed design created by Andy Newton from ARIN and his original design, which was called WHOIS was using XML and we in ICANN also did some prototypes and we were using XML, but when it got to the

ITF, the discussion there, the newest, hottest thing available in technical committee was JSON, and so it was decided to use JSON. I don't think a very good explanation, it's just it was what was decided to use.

UNKNOWN SPEAKER: So, you're saying that ICANN just adopted since it was the RDAP standard used JSON, so ICANN just ahead with it. That is what you're saying?

FRANCISCO ARIAS: Well, it's not so much ICANN, that's how the protocol was standardized in the ITF. So we were implementing RDAP, the standard has to be implemented according to the standards in the ITF. So the ITF was the forum where it was decided to use JSON instead of XML. But in terms of which you're getting, there isn't much of a difference, you have to decide between using one or another. You could also have used another technology, I don't know, JMUL, I'm not even sure if I'm pronouncing correctly or, I don't know, CBOR, or something like that. It is just a matter of deciding you have to use one and that that was what was decided.

STEVE CONTE: I'll throw a question to you. So RDAP was being developed before GDPR came out. Was the onset of GDPR, did it cause much change

within the development of RDAP through the ITF process? Were there many things that had to change within RDAP because of GDPR?

FRANCISCO ARIAS: Good question. I don't think there was anything that was added to the protocol as a result of GDPR. I think when the protocol was being designed in the ITF there was already the thought that there was a need to have these type of things take it into account. As a matter of fact, I'm no expert on GDPR at all, but my understanding is GDPR or something like that, privacy rules, were already in place for quite a while. What changed in 2017 was the penalties that companies could be subject to, or something like that.

But privacy rules were for quite a while, so they were already around at the time the protocol was defined and they were kept in mind so that you will enable to have things like differentiated access and, for example, the protocol does not get into what fields should be returned or not, it leaves that completely to the policy decisions. So in that sense you could say is flexible on what policies you implement; it was something that was kept in mind while designing the RDAP protocol.

STEVE CONTE: Thank you. Any other questions? Yes.

UNKNOWN SPEAKER: Hi, my question is on the user; where I'm from, registrars use, normally still use WHOIS on checking the information. What can I do to make

them, I don't know, be comfortable in using RDAP? Because what I see from the protocol itself is the same as WHOIS, and the information redacted from the registry side are still redacted on the RDAP side. So what they use on the registrar site if they have to adapt using RDAP as a protocol instead of WHOIS?

FRANCISCO ARIAS: Thank you. So, on the utility space, the registrars like the registries, they are required to offer an RDAP service starting a couple months ago, and many of them are already offering it. I think what we're seeing at the moment is the natural start of things, it will be a slow rollout of RDAP as people get to know that this thing exists and this is what is coming, it has better features that will enable people to do things they could not do in WHOIS. So I think it's just a matter of time that we could start seeing people referring to RDAP and using it more, but like said, from the implementation side, most of the registries and registrars in the utility space are already offering an RDAP service.

STEVE CONTE: I'm not seeing any more hands; ah, of course, the other side of the room.

UNKNOWN SPEAKER: So, I realized JSON it's an acronym for JavaScript, and you can correct, I forgot the rest of it already, just shows you my lack of memory, like a goldfish. But either way, I'm curious to know what

other programming languages are there besides JavaScript that are very prominent in RDAP and whether it's useful just to learn JavaScript versus learning any other necessary programming language that you think is necessary.

FRANCISCO ARIAS: JSON stands for JavaScript Object Notation. That's the acronym. And I should clarify that it doesn't mean that there is JavaScript programming in both. It's just the name of the standard, in regards to their presentation, I believe their presentation has some resemblance to how data is represented in JavaScript, that's the reason for the name, that's my understanding, but there is no JavaScript involved in RDAP. And probably what made it more confusing, is the fact that I said that our web client is implemented in JavaScript.

But you could implement an RDAP client in any language, or I guess I should say most languages, I don't know if every language. For example, the command line implementation that I show from ARIN, that's implemented in Ruby, I know their language, but you could do it in any of your favorite programming languages as long as they support doing HTTP queries, you are probably fine with it. So there is there is no requirement for a specific programming language in order to use RDAP.

STEVE CONTE: I want to make sure I get my steps in, so I've got another question over here.

UNKNOWN SPEAKER: Thank you, Richard Lau from Canada. Are there any examples of the object search. So if I'm understanding correctly, I could search for a name server and have a list of all the names that are on that name server or can I search for a registrant company name and have a list of all the domain names that that registrant is assigned to in the RDAP database rather than the WHOIS database?

FRANCISCO ARIAS: I have to confess I have not looked for registries. I don't know if registries have implemented this functionality, so I'm not aware of anyone, but I haven't looked for it, so I don't know if anyone has implemented this functionality.

STEVE CONTE: I'm seeing no hands. Going once, twice. Francisco, thank you so much for giving us the rundown on RDAP. Please everyone, Francisco Arias. So, Francisco's slides are already on the meeting schedule site or the meeting site, because there was a live demo those are clearly not on the slides, but there will be a recording of this session done and that will be on the site as well in the near future, I'm not sure what the turnaround time on that is. So if you wish to relive the live demo, just check in within a day or so, and it'll probably be up on the site, as well.

We are also running the session again on Tuesday, I have it, here, Tuesday at 3:15 pm, so if you missed some of this or you want to recap

the recap, please join us again on Tuesday. In the meantime, as far as how it works for today, we're going to give you a short break or a pretty long break, actually, so enjoy lunch. Our next session will be in the main room, in Room 517D as in David, which is right next to the escalators, it says GAC Room right now. That will be the root server operators we'll be discussing the root server system.

And then after that, at 5 pm we'll have ARIN which is the regional internet registry for North America talking about RIR activities. So please continue to join me throughout the day. And in the meantime, thank you for giving me your time and joining these sessions. Thank you.

[END OF TRANSCRIPTION]