KOBE – How It Works: RDAP
Sunday, March 10, 2019 – 08:45 to 10:15 JST
ICANN64 | Kobe, Japan

CATHY PETERSEN Good morning, again, everyone. Welcome to our How it Works tutorial today. This session, we will be talking about registration data access protocol, or what we call RDAP. Our presenter is Gustavo Lozano from ICANN's GDD technical services. Gustavo?

GUSTAVO LOZANO: Okay, thank you. Can you hear me? Okay, let's start. Before anything else, thank you so much for being here so early in the morning. So, the title of this session is How it Works and I'm going to discuss about RDAP, this new protocol that hopefully you can hear about, and it's like the new thing, right, like it's right now being implemented on ICANN. So, this is the agenda for today's session. First, it's going to be the introduction about what is RDAP, how it works and all of that, and then we are going to discuss about the RDAP basics, queries and responses, features and concepts that we need to understand in order to be able to read all of the documents about RDAP that hopefully you have been seeing around. I'm going to talk about differentiated access and the future and ongoing work.

So, before we can start talking about RDAP we need to understand why RDAP was developed, right? And RDAP was developed as a replacement for a protocol, a protocol that you are familiar with which is called WHOIS, and WHOIS is very old protocol which was developed, I don't know, 20 or 30 years ago, a long time ago, and it was developed when the Internet was not as complicated as today, let's say, so it has a lot of issues. What are those issues? The first one is there is no standardization on the format of the output. So, if you go to a WHOIS server in Brazil or in Mexico, or anywhere else, you will find that the response is different between all those different servers, and it's difficult to understand where the exact pieces of information are located in that response. There is no support for internationalization, so many of you that have worked with WHOIS servers, for example in China or Japan, sometimes you receive these weird characters in the WHOIS output, but it's not like weird characters, right? In reality, those are code points outside of the Latin script, but unfortunately because we don't have internationalization in WHOIS, our computer is not able to display those characters as they are supposed to be displayed.

There is no way to authenticate or provide different outputs, depending on the user, and this is really important because this is basically the killer application for RDAP right now. When the IETF develops a new protocol it's really difficult to get that

protocol implemented around the world, and obviously RDAP is going to be a global protocol that's going to be used by different parties and different clients around the world, so it's really complicated to get that new protocol supported. But in the case of RDAP, we have a clear application and that clear application is that with GDPR we need to provide different outputs based on the type of end-user that is asking for that information, and if that user is authenticated or not.

So, this is really important. This is, right now, the killer application for RDAP, and hopefully this killer application is going to allow us to implement the protocol quickly. WHOIS is lookup-only, and so there is no support for the servers. For example, if you want to search all the domain names that start with "icann" or something like that, that's not possible. There is no way to redirect from a registry, for example, to a registrar in order to provide extra information to the end-user. There is no standard way to know which is the server that I need to query to get a response. What is this?

Those of you that have been following the new gTLD program, in the new gTLD registries agreement, we tried to solve this issue in WHOIS by saying that if you go to WHOIS, that [nick], that TLD, that's the WHOIS server in which you can find information for that TLD, but for the latest TLDs that is not the case. But in RDAP, there is a new way to find that server for that particular TLD. It's called

bootstrap and it's one of the really, really cool features of the protocol, and it's obviously insecure. There is no way to authenticate a server and there is no way to encrypt the data between the server and the client.

This means that if you go to the Internet and you do a WHOIS query, anyone that is able to, let's say, listen on the channel is going to be able to get, or to see, the query and the response. So, these are the issues with WHOIS. As I told you before, WHOIS is really old, and the idea is that RDAP is going to replace this protocol. So, what is the timeline of all of these efforts to get into RDAP? Well, back in 2011, there was an SSAC report, SAC 051, which basically said that the ICANN community should work on a replacement for WHOIS. That was basically the guidance that we received from the SSAC.

So, in 2011, the board adopted that report, the SAC 051 and in June there was a roadmap to implement RDAP. So, at that point in time, the IETF started to work on that replacement model. In 2015 the RFCs, or the standards were published, and on June 15th they were beginning to have what we call the gTLD RDAP profile.

I want to talk more about this profile in the next slides, but it's a real important document that we need to have in order to get interoperability between implementations. In 2016, the first version of a gTLD RDAP profile was published. That was Version

1.0, and in 2016 in August, the Registry Stakeholder Group, they submitted a Request for Reconsideration regarding a policy called Consistency, Labeling and Display, because at that point in time the first version of that policy included a requirement to implement RDAP. So, in 2017 ICANN published a new version of that policy that removed that requirement.

In May 2015, the Temporary Specification for gTLD Registration called for an implementation of RDAP, the definition of a profile and the definition of service-level requirements and monthly reports to help ICANN decide what is the volume of queries that are received from RDAP. In August 2017 ICANN received a proposal from thy Registry Stakeholder Group to start a pilot program with some implementations. In September 2017 ICANN responds to the Registry Stakeholder Group saying, yeah, we accept this proposal, and that's when the pilot started.

In 2018 in August a proposed gTLD profile that was developed by the contracted parties was published for public comments. In February 2019 ICANN issued a notice for the contracted parties requiring them to implement RDAP, and on that date the official profile was published, and this is a really important milestone because a profile is going to help us achieve interoperability between all those different implementations of RDAP servers and clients. The deadline to implement RDAP for gTLD registries and registrars is August 26 of this year. Any questions? No.

So, what are the features of RDAP? Well, the features of RDAP are really analogous to the limitations of WHOIS, right? So, for example, one of the features of RDAP is the standardization of queries and response messages. So, now, it doesn't matter which server you're going to contact to get information, you will always be able to parse information and understand that in that specific field you will find a name, or the city or the country, or whatever information you are looking for.

The access to RDAP is secure. In the case of the TLD space it's going to be required to be under HTTPS which is the secure protocol under which the Internet basically runs right now, I mean, for security. Protocol can be extended. Those of you that have worked with EPP, in the case of registries and registrars, it's the same concept. You can create extensions and those extensions can be used, as the name implies, to extend the protocol to implement new features. The protocol enables differentiated access. So, what that means, and that's really important right now because of GDPR, and so basically it means that if you are authenticated you may be authorized to get full access to restricted data instead of redacted data, as is the case now in WHOIS.

There is a bootstrap mechanism in RDAP. So, basically, what this helps us with is to find the RDAP server for a specific TLD. As I mentioned before, in the case of TLDs with WHOIS, it's not that

easy to find which information to send the query to, but in the case of RDAP that's automated. Next, in the standardization, there is a mechanism to pre-direct from a registry to a registrar. So, in the case of failed registries, you can go to the registry, get the information or the basic information for the domain name, and then you can go to the registrar based on the response and be redirected to that RDAP server of the registrar to get more information.

The protocol is built on HTTP, and HTTP is a well-known protocol. I mean, everybody has used HTTP in this room. It supports internationalization and enables searches for objects. For example, in RDAP you can search for a domain name that starts with ICANN or whatever you are looking for.

Implementation Status. What is the implementation status right now for RDAP? As I mentioned before, the temporary specification requires registries and registrars to implement RDAP, and the idea is to follow a common gTLD RDAP profile which was published already and requires those registries and registrars to agree to some service-level requirements, and it requires those registries and registrars to provide information to ICANN on the monthly reports.

What is this information in the monthly reports? For example, the number of queries that we receive from RDAP. At some point in

time WHOIS Port 43 is going to turned off. We don't know when, but that is the idea, and, obviously, one of those datapoints to help us to make that decision, or help the community make that decision, is going to be those monthly requirements, because those monthly requirements are going to tell us the number of queries that we're getting over RDAP versus the number queries over WHOIS, for example. When the communities realize that WHOIS is not used anymore, then that's the point in which a decision is going to be made to deprecate RDAP.

The ICANN organization continues to work with contracted parties, and that means the gTLD registries and registrars, on finalizing the service-level requirements and the requirements for the monthly reports, and so with that ongoing work, the profile was finalized. Questions? No questions? Yes.

DEAN MARKS: Hi. Dean Marks, I'm with the IPC and I'm not a technical person, I'm a lawyer, so if this question makes no sense, I apologize.

GUSTAVO LOZANO: No worries.

DEAN MARKS: Does RDAP also facilitate in its search function what is sort of commonly known as a reverse WHOIS lookup, where it's not lookup based on common elements of a domain name, but a

reverse WHOIS lookup based on the elements of the WHOIS data itself? Thank you very much.

GUSTAVO LOZANO: So, that's not a feature of the core protocol. So, if you go and read the standards that are published right now, that feature is not there, but there is an Internet draft which is the way for the IETF to start working on something new, and that Internet draft from a colleague from Italy is going to add that feature to RDAP. Obviously, at some point the standard is going to be finalized to allow reverse searching. That doesn't mean that it's going to be required for gTLD registries and registrars. For that, some policy should be developed, but at least from the technology perspective, the feature is going to be developed at some point. In fact, that specific feature is one of the feature that the working group in IETF is working on right now, on reverse search. Questions?

So, Basics for RDAP, and, sorry, if I get too technical, yes, tell me that's yes, too technical, okay? So, RDAP was developed using what we call a RESTful architecture. A RESTful architecture is a common way to define APIs. What is an API? When you develop a software that is going to communicate over the Internet to execute some kind of process that is going to be automated,

that's when you use an API, and API stands for Advance Programming Interface.

This RESTful architecture is really common right now on the Internet. When you develop some software that is going to interact with, let's say, Google, Amazon and those kinds of IT companies, it's likely that you are going to use a RESTful API. A RESTful API takes advantage of HTTPS protocol. So, all the libraries that are already developed to handle HTTP are able to work with RESTful APIs. So, RDAP as I mentioned was developed as a RESTful API, and that's a real important distinction for RDAP, that it's a RESTful API. As I mentioned before, it uses the HTTP protocol. So, what does this mean? It means that the status codes, the verbs that we use in HTTP, those can be used for RDAP. So, for example, I'm pretty sure that a lot of you have seen the 404 on your browser, right, which means the place doesn't exist. So, in the case of RDAP, what do you think that 404 means? It means that domain name doesn't exist and if I'm looking for a domain name, and I get a 404, it means that the domain name doesn't exist.

So, this is how it works. Basically, it uses the HTTP protocol as the basis and it's important to mention that RDAP is read-only. This means that you can only perform read-only operations. It could be potentially extended to support all kinds of services, but for now it's read-only, and the only methods that are supported are

[GET and HEAD.] Again, sorry if I get too technical into this, but it's important to understand this.

Also, RDAP supports two types of queries. One is lookups and one is searches. Lookup will basically find information or exact-match information for a particular object. So, if you send a lookup for "icann.org" you will only get information for icann.org, for that specific domain name. And, on a search, you can tell the server, hey, I want to get information of all of the domain names that start with a certain pattern or contain certain pieces of a string, right? So, for example, with a search you can perform a search for all the domain names that start with "icann" and as I mentioned before there is some ongoing work to support reverse search. It's not there in the core protocol, but it's going to be standardized at some point.

So, what are the objects that you can query on RDAP right now, or basically what are the lookup paths that you can define and that are defined in the specification? Using RDAP you can get information for domain names like icann.org, like make that example. For the nameservers or the servers used for the delegation of the domain name, you can also get information for an entity. An entity is what you call contacts in WHOIS. So, the contacts in WHOIS are now called entities. You can get information for IP addresses. This is used by the regional Internet registries like ARIN, APNIC right, those guys. You can also get

information for ASNs, or autonomous system numbers. Again, this is not for the domain name registries. This is more for IP registries, and there is also a part called “help,” which I'm pretty sure it is easy to understand that is for getting help from [the server.] So, these are the types of objects that are supported in the protocol, and the following search paths are defined in the specifications. So, what is a search path? A search path, as I mentioned before, it's the possibility to search based on certain patterns.

So, for example, for domain names you can look for domain names based on the names, so you can search for the domain names that start with "icann" and you can search for domain names which are delegated to certain nameservers. So, for example, you can search, hey, I want to know all the domain names that are delegated to this nameserver, "ns.example.com" or you can search for domain names that are delegated to nameservers that contain certain IP addresses. So, you can search for the domain names which the domain servers contain certain IP addresses. You also have searches based on nameservers and entities. For example, you can search for entities which names start with registrar XYZ or registry Verisign, or whatever information you're looking for in the name, you can do a search for that particular piece of information. So, these are the lookup paths. These are all the objects that you can perform

lookups in RDAP, and these are all the search paths, or these are all the search functionalities that we have right now in RDAP.

Now, it's important to understand that one basis for RDAP is JSON. So, what is JSON? As I was mentioning before, the idea with RDAP is that the servers and clients can interact and parse the information without the need of a human being. So, basically, computers can process this information in an automated way. In order to achieve that kind of automation you need a format that is going to allow you to represent the structured data. So, this format in RDAP is called JSON is well-used in the industry. You have maybe heard of XML, so JSON is like XML, it is a way to represent [structured] data.

So, in JSON which is also specified in an IETF standard, we have the following types of values. We have numbers, strings, arrays, objects, and I apologize if this is too technical, but it's important to mention this because it's the basis of the protocol. At this point what is important to understand is that there is a format which is called JSON and this format allows computers to interact between them, and in the case of RDAP we use JSON for that response. So, if you get a response from an RDAP server you will get a JSON response. One of the cool features about JSON is that it's easy to read for a human.

So, this is JSON, what you have on your screen right now, and this is basically trying to define the data field for a book. So, for example, this book is called *Introduction to JSON*, there is some author, the number of pages, if it's published or not, and you can see the table of contents. So, you can read this, right, and you can understand that this a definition of a book, but also this can be parsed by as computer and this is the cool part of JSON. We can read it, and also the computers can easily parse this information. So, if a computer gets a different book it understands that if it reads the title that means the title of the book. If it goes and gets information for the number of pages, which is in the pages field, it's the number of pages. So, that's the really cool feature about JSON, that computers and human beings can easily read this information. Questions?

UNIDENTIFIED MALE: Sorry, going back to the search part, you said that you could search on AS numbers. Is there a limit to the return you'll get back on that? Because, obviously, you could have a lot of domains on a single AS.

GUSTAVO LOZANO: Yeah, that's one of the issues. So, the community, right now gTLD registries and registrars, they are discussing how to support search, or not. Right now, in the profile it is not supported. There

is only one search path that is supported which is looking for nameservers based on the IP address, but the other search paths are not supported because of what you're saying, right? You can do a search for, I don't know, "abc*" or a wildcard, and you will get millions and millions of domain names. So, that's going to be really difficult for servers and for clients to handle that kind of size. So, there is still some work going on, on how to implement searches, because of that limitation, obviously.

UNIDENTIFIED MALE: Okay, thank you.

GUSTAVO LOZANO: Yeah, so the only one that has been defined for gTLD registrars and registries in the profile is nameservers searched by IP address. That's the only one.

CATHY PETERSEN: Your name and affiliation, please, thank you.

YORAM HACOHEN: Yoram Hacohen, CEO of ISOC-IL. Does RDAP support wildcard search?

GUSTAVO LOZANO: Wildcard, you mean?

YORAM HACOHEN: Yes.

GUSTAVO LOZANO: Wildcard, but I mean for the search patterns, if you want to use search, you use the wildcard as a way to specify. For example, the domain names that start with "icann," right, so you specify "Icann*".

YORAM HACOHEN: Yeah, but can you have, instead of the star, if I want to have just one character in the beginning, so is it a rich support of wildcard?

GUSTAVO LOZANO: It's a limited support of wildcard. It's basically wildcard can only appear once in the search segment, so it's very limited what you can do with search. And, as I what you can do with search, and as I mentioned before.

YORAM HACOHEN: And it can appear also in the center of the string? So, we're just starting with the –

GUSTAVO LOZANO: So, the example in the protocol that we have in the specifications is something that starts with some string, and then you can use the wildcard at the end, but I'm pretty sure that you can use it at the beginning. But I will review the specification again and get back to you. So, search is something that's still being worked on in the IETF. Questions?

So, now let's go deep into the protocol. This is going to get somewhat technical, so please try not to sleep. So, we need to understand the common instructions for RDAP, right?

So, in RDAP there are some Common Data Structures that are going to be used in all the responses or in some of the responses, and the first one is the Object Class Name. So, the Object Class Name identifies the type of object that is being processed in the response, and we have five. The domain name, nameservers, entities or contacts, IP networks and autonomous system numbers. So, the domain names, you are already familiar with those. You know what, obviously, domain names is, right? The nameserver, that's probably something that you already have been working with in your different positions. Entities equals contacts. The IP networks, if you are familiar with IP registries, like ARIN, APNIC, those are the prefixes that are assigned to end-users or to corporations. And autonomous system numbers or [ASNs.] These are used in the routing protocol called BGP, so if you have worked in the routing area and know about BGP, this is

that the information about [ASNs.] But here in the domain industry, probably the first three are the only ones that are interesting for us. The other common data structure that we have in RDAP is the "RDAPConformance" object, and this is really important because, as I mentioned before, RDAP is going to be extended, right? And like with EPP, we expect to have several extensions. Why? Because different registries have like secret ways of doing things, or "sauces" to solve problems, so they implement their own extensions.

So, the RDAPConformance object is a way to define which are the extensions that these responses comply with. So, for example, if in my registry I have this special object because I need to do something special about the city, or whatever, and I define an extension, I can specify in this object, hey, this response complies with this specific [string.]

Links. Links is another data structure that is using RDAP, and basically the idea of a link is to provide a link to a resource that is related to the response. So, what does this mean? For example, the second link that you have on your screen, or, oh well, maybe it's easier to just highlight it, but for example this link will be used by a registry to provide a pointer to the RDAP server of as registrar in which more information can be found. So, a link is basically, yes, a way to reference a resource that is related to that domain

name, or nameserver, or a response in RDAP in which the client can get more information.

Notices and Remarks. A notice can only appear once in the main object. It's an array, so what that means is that it can have several notices within the same object. Again, this is maybe too technical, so I apologize. So, this is the object that I am talking about. This is a notice. So, in that notice, remember, you can have several notices. For example, these notices about object truncation. Due to certain laws or regulations, I can have another notice regarding if you want to complain about the data in this domain name, you can go to this page or whatever.

So, you can have several notices within the response, and the ideas of these notices is to provide a human being with some information that is important to understand that response, and we also have remarks. Remarks are like notices, but they can appear multiple times within the response and those are meant to provide information for a certain object. So, I can have a remark for a nameserver like this nameserver has been used one thousand times, so some information that I want to provide for that specific object.

The Language Identifier. Remember that I mentioned that in RDAP we have internationalization support. This is one of the features that allows us to have internationalization. For example,

if I want to display an error message, domain name not found, I can specify that the language is English. I can provide that information in a different language, and I can tag the language of that message.

Events. In RDAP we have the notion of events. So, events are actions that have occurred on the object. So, for example, one event is when the domain name is going to expire which is this event that is highlighted. So, this means that this domain name will expire at this particular point in time. You can have an event for the registration of the domain name, the last time that it was updated and there are several events that have been defined in the standard protocol.

Status. Those of you that have worked with WHOIS, you have seen these domain statuses on the response. So, the domain statuses are basically the EPP statuses that are used by registries and registrars to provide the status of the domain name or the object. So, in RDAP we also have support for that. So, if you do a lookup for a domain name, you will get an array of those statuses. All the EPP statuses in EPP are not supported by default in the core specification, but there is an RFC, a new standard that was published by the IETF that maps all the EPP statuses into RDAP. So, all the statuses that you can have on WHOIS are now supported in RDAP. So, we have full interoperability in that regard. Another common data structure is port43 WHOIS and this

is used to provide the WHOIS server in which more information can be found. We expect that in the future this data structure is not going to be used anymore because at some point WHOIS 43 is going to die. But for now, you can specify in the RDAP response what is the WHOIS server.

Public IDs. This shows the public identifier of an object class. For example, in an RDAP response you can see the IANA registrar ID. For those of you that are in the gTLD space, you know that on IANA we have an identifier for each registrar, and that identifier is unique for that particular registrar. So, public IDs are a way to display, for example, that IANA registrar ID in the domain name response.

So, those are the common data structures. As I mentioned, these are used in all different kinds of responses. Questions? No.

So, now let's get into the Queries and Responses. So, remember that I mentioned that we have five object [classes or principal] object or main object classes in the protocol. So, those object classes map into the lookups that we can perform on RDAP. So, we have the domain name, nameserver, entity or contacts, IP network and autonomous system numbers. Those are the five things that we can look up in RDAP right now.

Domain query, pretty simple. Once you know the base URL of the RDAP server – and with base URL I mean something like

http://servername/somethingelse, when you have that base URL, then you just add slash domain, slash the name that you're looking for, and you'll get the information if the domain name exists. This lookup path supports [LDH,] like domain.example, supports IDNs in U-label format like the second example, and IDNs in A-label format like the third example that you have on your screen. Again, we have full support for internationalization. This was not possible in WHOIS for example and it's now possible in RDAP to do these kind of queries. Questions?

And when you get a response for a domain name, these are the fields that you get in the response. The first field is the handle. This is the unique identifier. If you have used WHOIS, this is like the [inaudible] that sometimes you see in EPP or in WHOIS. We have the LDH name. This is the domain name in LDH form, so basically, letters, hyphen and numbers.

We have the Unicode name, domain name in U-labels. In case of an IDN, you will find a U-label in that element. We have [variant number,] and this is also a new feature on RDAP. On RDAP, we have a way to specify which are the variants for that particular name. So if you have café with an [ASN or café with or without a VSN,] you will see those variants there, if the registry supports variants, obviously.

Nameservers in which the domain name is delegated, entities or contacts, you can put information about the network of the nameserver, and we have DNSSEC information. So the same elements that are transferred between registries and registrars for DNSSEC, you can also provide that information on RDAP. Questions?

Basically, what we have in WHOIS was translated into RDAP, into JSON, so all the things, all the elements that you can get in a WHOIS response, you can also get it on RDAP, plus several more, like variants for example.

Nameservers. Nameserver or nameserver queries, those are used to get information about a nameserver, like "ns1.nic" for example or "ns.example.com. So, if you know the name of the nameserver, you can do a query, using the base Url/nameserver/nameserver name, and again it supports nameserver in an LDH format, like the first one, and also A-labels and U-labels. So, you have examples of all of those in this slide. This is LDH, this is an A-label, and this is a U-label. Questions?

These are all the elements on the nameserver response. Again, we have a handle, we have the name of the nameserver in LDH format, we have the name of nameserver in Unicode, and that's the Unicode name, and we have the IP addresses of the nameserver. We can use IPv6 or IPv4 in the protocol, both are

supported. We have entities. Those are the contacts in WHOIS. This is the entity query. We can do a lookup for a contact. Probably with GDPR this is not that useful anymore, but once we have differentiated access, you should be able to get information for a contact using this lookup path. So, you have the <baseUrl>/entity/<handle>, and in that handle you provide a unique ID of that contact, that contact ID in WHOIS terms or EPP and you will get information for that contact. This is an example of the handles that you may be familiar with.

UNIDENTIFIED MALE: In the entity query, do you support IDN?

GUSTAVO LOZANO: Yeah.

UNIDENTIFIED MALE: Yeah, so the examples are also that it is also possible to search with a different language then.

GUSTAVO LOZANO: Yeah, but this functionality is meant to use the handle, right? So, normally, handles are specified in LDH, like letter, hyphens, numbers. But if a handle is defined using code points outside of the Latin repertoire, yes, you can use those. For example, you

have a handle with Kanji in Japanese, and then you will, you can use it.

So, this is information that you get for an entity. You have, again, as I mentioned before, the handle. You have a [bigger] array and I will enter into those details in the following slides. You have roles. That's the relationship of the object with some other entity. For example, you can specify that this entity has the role of a registrar, and within that entity, you can have entities with the role of the technical contact or admin contact. So, you can have entities within entities, for example, and you use the roles to specify what is the role that the entity is playing for that specific response.

You have events, and again for the IP registries you can have networks and autonomous system numbers. But I think that the most important element right now is the vCard because in JSON, on all the contact information is displayed using jCard, and let's see what is that?

So, vCard and jCard. vCard is the detailed representation of a business card. This is an example of a vCard. This is used all over the place on the Internet. So, you have Outlook, for example, and you export your contact information. That information is going to be exported as a vCard, or at least you have that functionality to export it as a vCard. Again, this is in the standard for [the

specificaiton,] and that's the RFC6350, and this is what we use on RDAP, but you remember that RDAP is defined in JSON. Because of that, the vCard was defined on JSON and the specification is called jCard.

So, jCard is just the representation of the vCard in JSON, and this is an example of a jCard. This is what you will get on RDAP for any specific contact. So, you have the full name, you have the name of the user, what kind of user is this person, and in this case it's an individual. You have the address. You have a telephone number and you have an email. And this is the way that a jCard is mapped into a vCard. For example, the address which is something that a lot of people using WHOIS or RDAP may be interested in, and it's something like this. And this is [in the structure ordered] data.

So, the first element is the post office box. The second one is the extended address. This is the street. This is the city. This is the state or province. This is the postal code, and this is the country name. It's important to mention that on RDAP what is specified is the country name, and then the two-character ISO code, like in WHOIS. Right now, there is an extension that has been proposed for RDAP to support the two-letter code from the ISO code. So, right now, if you go to RDAP, you don't get like, I don't know, "be" for Belgium or "mx" for Mexico, or "br" for Brazil. You get the full name of the country. That's going to be extended.

JUAN: I'm going to [make] my question in Spanish, is that okay? [in Spanish/no translator].

CATHY PETERSEN: Can you translate that into English, please?

GUSTAVO LOZANO: Yes, I can translate the question in English. So, what he's asking is, when you go to RDAP and you query for, let's say an entity, right, and that information is going to be redacted, that's the term in RDAP. If there isn't a special [HTTP] status code indicated, then the response was redacted.

JUAN: Not if there is any special response, but rather if there is a common response that we are going to get if someone, some other country implements something similar to RDAP due to GDPR.

GUSTAVO LOZANO: Yeah, so the answer is yes. There isn't a special response in RDAP to indicate that that response was redacted, and not only the response, but it goes into more detail than that. You can specify that a certain element was redacted. For example, maybe just the

telephone number or just the email, so you can go to that level of detail, but yeah there is a way to specify that. I mean, you can specify whatever. There is, I have an example, so I will show you an example.

JUAN: Okay. Great, thank you.

GUSTAVO LOZANO: I think it's just easier to see it in the example than trying to explain, but yeah there is a way to define that and to specify that. So, this is not for the domain name industry, but if you also have knowledge of the IP registry industry, these are useful for that kind of industry. So, the IP lookup allows you to query for the specific IP address. So, you specify the IP address and you will get the response for the most specific prefix covering that IP address. So, if you have it there as "/24" and you delegate it and it's "/27" from that "24" you will information for the "/27." No worries. If you are not in to the IP registry, this maybe doesn't make any sense, but this is for entities like APNIC, ARIN, and those kind of industries, and you can do queries for IP addresses. For example, you can query for a specific IP address, or you can go and query for a prefix like in the "/24." And this is information that you will get for the IP address. You get a handle, as with all the responses. You get the first and final address for that specific [look.] You get

the version, which can be v4 or v6, and other elements for that response.

Autonomous System Numbers. On the Internet, there is a protocol called BGP. That protocol is used to interconnect all these different networks around the world. So, if you have an ISP and you have a second ISP, on the Internet they will interconnect using BGP. So, in BGP, you have the concept of an autonomous system number. That autonomous system number identifies a particular network. So, if you are in the IP registry industry, you are interested in getting information from a specific autonomous system number and this is the way to get that information. You specify the ASN, like 651 or 327, and you will get information for that network. And this is the response that you get for an autonomous system number query.

And finally, we have the "/help" which the name implies is to get some help from the server. This is an example of a help response. And finally, we have Error Responses, and an error response is basically to indicate that there is an error. For example, if you get a 404, what it means is that the domain name doesn't exist, if you're looking for domain names. The error code that you see in this field is based on the HTTP status code of the header of that response. So, if in the header you have 404, this is a way to indicate more information about that 404. Questions? This is the technical portion, so, yeah?

CATHRIN BAUER-BULST: Good morning. I'm Cathrin Bauer-Bulst, I'm with the GAC Public Safety Working Group, work for the European Commission, and I just have a practical question because, as I understand now, there are a number of questions that are still being looked at at the moment, in particular when it comes to lookup and search, and I was just wondering how that squares with the requirement to implement this by August for the contracted parties, and how exactly this will work. So, is this sort of ongoing now with a view to then being implemented as part of this implementation by August, or would this be a later step in time? How do you envisage this practically working out?

GUSTAVO LOZANO: So, what I explained in all of these slides are all the features in the protocol. That doesn't mean that the registries and the registrars in the gTLD space need implement to all those features. In the case of the gTLD space, what the profile says, because we have a final profile that was defined by registries and registrars, it's that you need to implement domain name queries, the nameserver queries, the queries for entities, and these are the lookups, so these are the WHOIS lookups that you support right now. And in the case of search, the only search that needs to be supported is the search for nameservers based on IP address, so basically this

one. Those are the only things that are required right now in the profile. In the future, policy may decide that something else needs to be supported, but as of now that's what the profile says, basically.

CATHRIN BAUER-BULST: Okay, thank you. And may I just follow up with one more question? You showed an example of, I think it was a remark or a comment that can be shown, for example, for redacted information. And the gentleman before me already asked to what extent are responses standardized, and I was just wondering whether any thinking has gone into standardizing any of the content that would be returned so that, for example, if information is redacted, there are certain elements that should be included in this comment section that I understand is possible to attach to the response, such as the possible contact information for further follow-up as you were showing in your example. Has any thought gone into that?

GUSTAVO LOZANO: Yeah, that's going to be defined by policy, and it's defined by policy and basically what the gTLD RDAP profile is doing is, yes, mapping that policy into technical terms. So, in the profile, we need to show these, but the information that needs to be shown is based on what the policy says.

CATHRIN BAUER-BULST: Okay, thank you.

GUSTAVO LOZANO: Yeah. Yeah?

DEAN MARKS: Thank you. In terms of the entity portion of it, if a registration is making use of a privacy proxy service, is there a standardized response when there's an RDAP inquiry as to returning the information that a privacy proxy has used?

GUSTAVO LOZANO: So, the protocol supports to put any information into any of those elements, whatever the policy says is what we are going to implement using the profile.

DEAN MARKS: Okay, so it's very flexible.

GUSTAVO LOZANO: Yeah, it's very flexible.

DEAN MARKS: Okay, thank you.

GUSTAVO LOZANO: Sure. Other Features and Concepts. As I mentioned before, RDAP was defined with extensibility in mind, and this is supported, and I mean extensibility is baked into the protocol. You can extend the objects, you can extend the functionality of the protocol itself, and several IANA registries have been defined to extend the values and the protocol. And if you're an implementer of an RDAP client or an RDAP server, this is really important. If you extend the protocol, please go to these IANA registries and add a reference to that extension. What happened in the past with EPP analytic protocols is that we have all these different extensions and there is no central repository of extensions for an implementer to understand how to interact with all these registries and registrars.

So, the idea with RDAP is that it is really easy to go to this IANA registry and say, "Hey, I'm the registry for country *x*, or the registry for this region, or whatever, and I define this extension, and so please add this reference to the IANA registry so that other implementers know how my extension works." It's fairly easy. You just send an email to IANA, and there is going to be an expert review process, but it is a really lightweight process. The expert will read the extension, will look at it, and he will then proceed to publish that into the IANA registry.

So, these are the types that can be extended fairly easy by an expert review process. So, remember that we have the notice and remarks types. We have a status. We have events. We have roles. All of those can be extended. For example, if you are in, I don't know, New York, and the registry for New York, or I mean NYC, requires some special contact, that is called a nexus contact, for example, they can define a new role called "nexus contact" for that purpose, and they can extend the protocol to support that kind of contact that is specific to that city. They can go to IANA and add a reference there saying, “Hey, if you see the role ‘nexus,’ that means that it is a nexus contact for New York City,” for example. So, this is something that is really important to keep updated because if we maintain this registry as updated, implementers will have the possibility to implement all those extensions.

These are for all the different types, and this is the link for the IANA registry. So, if I click on the link, you will see that all these types have been defined, and you will see that this list just keeps growing and growing because all of the different extensions are being referenced from this registry. If you extend the protocol to provide more functionality that [isn't] specified in the core, you can add that reference to this registry on IANA and this registry basically has a reference to the specific extensions that have been developed by other entities. For example, this extension is from

ARIN, and if you go to the reference, hopefully, you will find how you can interact with a registry that implements this extension.

Bootstrapping, and this is a really cool feature of RDAP which was not available on WHOIS. On WHOIS it was kind of complicated to get the WHOIS server of the TLD that you were trying to get information from. In the new TLD we tried to solve this issue in the new TLD agreement by saying if you have a TLD example then the WHOIS server should be WHOIS.nic.example, but for legacy TLDs, that may not be the case.

On RDAP it's pretty simple. If you have a client, it will go to this address which is an IANA registry, and this address contains this registry which basically says if you want to get information, for example, for a domain name that ends on Bs, or Cr, or whatever, you go this base URL. So, if I go to this URL, this is what the client is going to do. Base URL, the domain name, and then I will get the RDAP response for that name. That's the RDAP response for nic.br, and I get the base URL from this registry, and so for a client it's pretty simple. They go to this address, get this JSON file which contains all the TLDs with all the base URLs and then it will just create the query like what I just did. That's the bootstrap mechanism.

References from Registries to Registrars. This is also one of the cool features that we have in the profile. In WHOIS it was really

difficult to say from the registry, “Hey, if you want to get more information about this domain name, go to this WHOIS server of the registrar and get the information.” This is standardized on RDAP, so there is a way for the registry to indicate the registrar base URL to get further information about the domain name.

And this is something really, really new, the RDAP Pilot Working Group which is basically gTLD registries and registrars. They asked ICANN to create a temporary central repository of RDAP base URLs for registrars. So, in the gTLD space, when you get information for a domain name, that registry will use this bootstrap to tell the client, “Hey, if you want to get more information for ‘example.com,’ then you can go to this registrar and get more information from that registrar.” So, everything is going to be automated on RDAP and this is going to be used by registries to populate that field for that registrar.

RDAP Object Tagging. So, as I mentioned before, there is a way to do a bootstrap for domain names, for IP addresses, for autonomous system numbers, but there is no way to do that kind of bootstrap for an entity. So, imagine if you know that the handle for my contact is, I don't know, GUSTAVO123, there is a way now on RDAP to have a universal unique identifier across the domain space for my contact. So, it doesn't matter if you're looking for the domain name in a dot-com or a dot-net, you can get information from a contact if you implement or the client and the

server implements an object tie-in. And this is specified on this RFC 8251, and this pretty simple. You have the handle of the entity, you add a hyphen and a new unique IANA global identifier for that RDAP authoritative source. So, if my contact ID is GUSTAVO123, and that's under the com namespace, you may be able to add a hyphen and Verisign maybe, maybe if that's the identifier, and that's going to be like my global identifier across the name space. And we have some entries in that registry. Questions?

Internationalization. As I mentioned before, on RDAP you can use internationalization on the queries and responses. So, you can query for a domain name using the U-label for example, and you can get a response with UTF-8 or whatever Unicode – sorry, the response always use UTF-8 so all the repertoire from Unicode is supported.

Rate Limiting. Right now, WHOIS, most of the registries, they apply some kind of rate limit. This is to prevent attacks or to mitigate attacks on the WHOIS service, but unfortunately on WHOIS there is no way, as a client, to know that you are being rate-limited. In the case of RDAP, because we are using HTTP, there is an HTTP status code 429 that specifies that you are being rate-limited. So, as an end-user you are going to be able to say, "Hey, I'm being rate-limited because maybe I'm doing too many queries." Questions? No?

Differentiated Access, and this is the killer application for RDAP. This is the reason why everybody is rushing to implement this new protocol. In order to understand differentiated access, we need to understand certain concepts. The first concept is authentication. Authentication in all these conversations that we're having about differentiated access means, yes, verifying the identity of a user process or a device, so that means that the system is going to authenticate that my username and password, for example, is correct. And that's it. Yes, verifying that I am who I am telling the system that I want to be identified as. Authorization, on the other hand, is the concept of denying or allowing a certain user that has been identified to access some resource. So, I think this is simpler to see in this slide. So, authentication, is “Are you really who you say you are?” And the authorization, it's basically the question, “Do you have permission to access that resource?” When we have differentiated access, you will be authenticated and then you will be authorized. And based on your authorization profile you may get access to certain fields or not of the RDAP response. For example, maybe the user EXAMPLE123 has access to see the complete address of the registrant [content.] So, that is authorization.

Another concept that is really important in the case of differentiated access is HTTPS. As I mentioned before, HTTPS

stands for Hyper Text Transfer Protocol, it's the secure operation of HTTP, and when you use the browser, if you see the padlock and it's green or closed or whatever, then it means that the HTTPS is being used. If HTTPS is used on RDAP to support cryptography so that the query and the response cannot be seen on the Internet.

So, as I mentioned before, on RDAP, we use HTTPS, and HTTPS uses TLS, which stands for Transport Layer Security. And when we use TLS, we rely on digital certificates to authenticate the server and, optionally, the client. So, when you are on the Internet, normally, you are interacting between servers and clients and we use certificates.

So, what is differentiated access on RDAP? Differentiated access refers to showing different subsets of data fields based on permissions, and you can see these as roles. [In mind] that you have a role to see, as I mentioned before, the address of a registrant. So, if you have that role assigned to your user then you will be able to see that specific subset of the information. The temporary specification, as you know, defines the minimum output and also requires providing access to further data on the basis of legitimate interest. So, right now, there is some work going on to define how differentiated access is going to be implemented on RDAP. ICANN has been working on an RDAP web client that we hope is [going to be used] in the future to get

information from RDAP. This is meant to be user-friendly, so I'm going to do a live demo, and hopefully it works.

So, you go to this address, and this is our RDAP client, and it's pretty simple. For example, if I specify this domain name, and can you see in the back of the room? No? Now? If I do a lookup you will get this response, right? And this client is beautifying the response, right? So, this response can be read by a human being, but if you want to see the actual RDAP response, you can click on this link and you will see the RDAP response. For example, we have what we have mentioned during this presentation, the object last name, handles, remarks, links, events and all of that.

As you see, this is the minimum output that you will list from the temporary specification, right? So, how differentiated access may work in the future, for example, this client supports certificates, so if you have a TLS client certificate in your browser, you can click on this combo box, click lookup, and the browser is telling me, hey, this server wants you to be identified, and so this client certificate allows it to identify me as an authorized user. If I click "okay" then the response that I get, it's a full registration data response, so you can see the contact information and registrant.

SYED IFTIKAR H. SHAH: We have a question online from a remote participant, Syed Iftikar H. Shah. Is there any digital certificate used for authentication?

GUSTAVO LOZANO: Yes, this is an example. This is an example of using what is called TLS client authentication. So, in this case, this browser, I have a personal certificate and that personal certificate was installed into this browser. So, if I go to the browser and I go to certificates, you can see that there is a certificate assigned to this browser. So, with this certificate, which is a client certificate, I can authenticate myself and get a full registration data response. So, this is one where you're using client certificates, yes?

CATHY PETERSEN: There is a continuation. Whether the CA of the digital certificate accredited by the national authority, or otherwise?

GUSTAVO LOZANO: That will be defined by policy. Whatever policy defines is what is going to be implemented for ICANN.

UNIDENTIFIED MALE: It is an extension to the remote question; can you inform specific policy for issuing those certificates with the level of entrance to

the RDAP? Like I want to have face-to-face authentication in order to get a specific access.

GUSTAVO LOZANO: The answer is, yes, and there is right now a group which is called the TSG. Maybe you are aware of this group, which is trying to define a way to implement the differentiated access on RDAP and, yes, that's going to be available. We can also use username and password. For example, if I want to get information for this domain name, if I do a lookup, this is not going to be authenticated. This is the minimum output that you'd expect from a registry, but if I do a lookup with open ID, which is another technology – this is the technology that has been proposed by the TSG to be implemented for the differentiated access. So, this is what you may expect to see in the future. You will click something like this, and then you need to specify an identifier. In this case, we are using Gmail as an authentication provider and I can't remember the email or the password. So, let me get that, but this is like when you are on a website and it says log in with Facebook or log in with Amazon, or log in with whatever, and this is basically what you can expect in the future with RDAP. You will have log in with some provider and that provider is going to authenticate the user. So, in this case, I'm going to use this account, which is on Gmail, and this is basically what you can expect in the future with differentiated access. Oops, what is the password? No worries,

this is a testing account for that because of this demo. And this is the RDAP response when I was authenticated, so this is a full RDAP response that contains the registrant information, technical contact, whatever, okay? So, in this client we have implemented two technologies, client certificates with TLS and username and password using OpenID. Questions?

Future and Ongoing Work. As I mentioned before, RDAP is a flexible protocol and in reality, RDAP is a protocol that is so flexible that we need some kind of guidance for the parties that are going to implement this protocol to get interoperability between the parties. That guidance is called the gTLD RDAP profile, and for the gTLDs we have a profile now. That profile was defined by the contracted parties. You can get the profile from this link and you will see two documents, the RDAP Technical Implementation Guide and the RDAP Response Profile. These are all the technical guidelines that contracting parties shall follow, and this is a mapping of the policies on the ICANN gTLD sphere into technical specifications. As I mentioned before, ICANN is still working with the contracted parties to finalize the service-level requirements and monthly reporting. Yes?

UNIDENTIFIED MALE: Are there any software tools that are available if I want to implement RDAP on the server side? Are there any vendors that offer solutions?

GUSTAVO LOZANO: Yes, there are open source implementations of servers and clients. I [have another slide] on that regard.

CATHY PETERSEN: We have another question from Syed Shah. Is this interface open for the global community for testing purposes?

GUSTAVO LOZANO: Yes. Yes, if you go to the link that is on this slide you will see some information about that. Also, ICANN has a website with pointers to public pilot work that is going on.

CATHY PETERSEN: Thank you.

GUSTAVO LOZANO: So, the EPDP, I'm pretty sure that you are familiar with the EPDP, and I don't need to go into that topic, but I mean this policy development process was started because of GDPR and there is a final report from the EPDP Working Group and the GNSO

approved this final report on March 4th, so that's I think important news.

The ICANN TSG. In December the ICANN organization launched the Technical Study Group on access to non-public registration data, and basically the idea is that this group is going to propose a model to the community that can be implemented to support differentiated access. That report was published three days ago and basically what it says is that OpenID Connect is going to be used as the technology to implement authentication and authorization, and it's basically what I showed you moments ago. I mean, this is the same technology that we are using right now.

The IETF, some of you might participate on the IETF, and the IETF is the Internet Engineering Taskforce. It's the organization that publishes basically the standards, [and these] standards are used by the Internet to get interoperability. There is a working group on the IETF. This working group is called the REGEXT, or the REGEXT Working Group, and this is the home of all the efforts that are going on to define the future for RDAP and EPP, and this is a list of, let's say, the most interesting drafts or specifications that have been defined, and one of those, based on a question that was asked, is the specification to support reverse search on RDAP. And for the gentleman that was asking about open source implementations, on this slide you can see at least two links for

servers and several client projects, and these are open source so you can use them on your implementation.

And that's the end of the presentation. Do you have any other questions? Do we have questions?

CATHY PETERSEN: No.

DEAN MARKS: Thanks. Is it possible to get access to the slides that you presented, and how and what is the way to do that? I always forget. Sorry. Thank you.

CATHY PETERSEN: The slides are already uploaded in the public schedule for this session. So, feel free to download them. The link to this recording will also be added to the public schedule within a week or so, so the recording for this will also be added.

DEAN MARKS: Thank you so much. This session was so informative. Thank you so much.

CATHY PETERSEN: You are very welcome, thank you. Another question? Your name and affiliation, please.

IRANGA KAHANGAMA: My name is Iranga Kahangama with Public Safety Working Group. I just had a general question. Can you speak a little bit about the ability to have confidentiality in logging records if you were someone like law enforcement doing queries over RDAP?

GUSTAVO LOZANO: The TSG is working on that, so in the final report from the TSG that's one of the requirements to support for law enforcement or for other entities.

IRANGA KAHANGAMA: Got it.

GUSTAVO LOZANO: So, it's there, and it was considered. Believe me, I was part of the conversation.

CATHY PETERSEN: Alright, if there are no other questions, I guess we will conclude this session. Thank you again. We have our next How it Works tutorial at 3:15 in the same room, and that will be with the root

server operators, so that's a very interesting session, and of course at 5:00 p.m. we have our How it Works on DNS abuse. So, have a great day, and I hope to see you later at 3:15. Thank you.

[END OF TRANSCRIPTION]