BARCELONA – DNSSEC Workshop (2 of 3)
Wednesday, October 24, 2018 – 10:30 to 12:00 CEST
ICANN63 | Barcelona, Spain

UNIDENTIFIED FEMALE: DNSSEC Workshop two of three, 10:30 to 12:00, October 24th, 2018, room 113.

UNIDENTIFIED MALE: Yes, Steve is on the phone.

JACQUES LATOUR: Alright, we're going to start pretty soon here. Wait a minute. No. Alright, good morning. I'm Jacques Latour with CIRA, and Steve Crocker is on the phone, and we're going to be moderating this panel. So the goal of this panel was to continue a discussion we had in Johannesburg, the ICANN 59 meeting. That was a policy meeting, and during the DNSSEC workshop, we look at the policy impact of CDS, CDNSKEY, and CSYNC.

And the main reason we looked at it is back then, that's last year, when we started to look at the idea of scanning an entire zone and picking up the CDS and importing those in the registry and doing all of that, a lot of people said, "You can't do that because it violates ICANN policies. And that was the main reason.

And then when we asked, "Okay, what are those policies that it's violating?" Nobody knew what they were. But 100%, they were being impacted. So after this meeting, we started to do research on which

policies might be impacted, are there any policies regarding DNSSEC, and when we look at the base registry agreement for ICANN for gTLDs, it says the registry operator shall accept public key material from child domain names in a secure manner according to industry best practice. So doesn't prescribe exactly the method that it has to be done.

So in terms of policy violation in that manner, the base registry agreement has nothing there preventing them from getting the DS from the child directly. In the RAA, the 2013 registrar accreditation agreement, in there, it said that the registrar will process DNSSEC information by relaying orders to add, remove or change information. So its to relay information, it's not to store or process or capture and all that. So in this view, we came to an agreement that there's no violation of agreement there, because it wasn't clearly prescribed exactly what was to be done.

So now, we're looking at – because it doesn't violate any agreement, we have multiple options, and Steve is going to talk about where we're at, all the choices we need to look at to move forward, and that's it. Steve?

RUSS MUNDY: Steve, did you hear us?

STEVE CROCKER: Can you hear me now? Okay. Thank you. I was on mute. Okay, thank you, Jacques, and thank you, everybody on this panel. My concern is making the whole thing work somehow. So I see a lot of initiative here on the CDS and CDNSKEY approach, and if that's the one that succeeds,

that's perfectly okay with me. But I haven't yet seen that there is agreement and broad acceptance on the gTLD registry side and the registrars. In listening to the various approaches, it seems to me they fit into a very simple model of three binary choices. And can I have the next slide, please?

JACQUES LATOUR: We seem to have an Adobe –

STEVE CROCKER: Sorry, I'm at some distance here.

UNIDENTIFIED MALE: Yeah, we're having a little bit of trouble with Adobe, Steve, so go ahead. We'll try to get the trouble taken care of.

STEVE CROCKER: Alright. Well, then I'm going to wing it, because I don't actually have the presentation in front of me. But it goes basically like this. With the CDS and CDNSKEY publishing, there's expectation that the registry is going to scan and pull the records. So that's the pull model. The complimentary idea is that the new records are pushed upward through some sort of interface.

So that's one binary choice, pull versus push. A second binary choice is quite obviously whether or not we're communicating DS records or new key and the DS record is computed on the basis of that key. And the

third binary choice is whether the role of the parent is taken by the registry or perhaps by the registrar. That is, one could imagine a registrar scans the child zone as opposed to the registry doing it, and then the registrar uses [inaudible] interface to push that upward to the registry.

I see the slides are up there at least. Let's go rapidly through them with the next slide. Are we there yet? So those are the three choices. All the choices will work. You can even do more than one. I've noted that SWITCH is doing both a pull, and a push and has priority for the push, if I have been paying attention correctly. Next slide.

Who gets to choose? Well, that's open for discussion, and the next slide, so this is me speaking to the presenters during this panel, which is not only be clear about what parts you're doing but also what parts need to be done by somebody else, and this question which was hanging around as to whether there is some impediment on the political or contractual side. if there is, let's bring it out to the surface and deal with it, and if there isn't, then we don't have to deal with that.

But if we have to be explicit about who third-party DNS operators are and have some sort of WHOIS information, I know that's a terrible idea people don't want to touch, but let's just go after whatever the necessary issues are. And with that, I turn things back over to Jacques.

JACQUES LATOUR: Steve, we just got the slides back. Cathy, can you go back to slide number three? Yeah, the pull, the three choices. I think it's important

for people to – maybe you want to just go over this one quickly with the text on the screen so we can get our mind in the right mindset. Well, Steve, can you cover the three different choices again quickly? I think you're on mute.

Because the key thing we need to talk about is that the issue we have is either we push or pull. We have KSK versus DS, and the parent is either the registrar or the registry, and that's what we need to address in this workshop, is ideally, we shouldn't have eight choices, we should have only a few that the industry's going to move towards. You can go.

MICHALE HAUSDING: Okay, for SWITCH, we had push model so far, but we see some difficulties in that because of the many systems involved, and that's why we think the pull model is more easy. But we still keep the push available and give it priority over the pull, so in case you really intentionally want to change something in the parent zone, you still have the –

JACQUES LATOUR: [Do you have slides?]

MICHALE HAUSDING: Yeah, I have my slides. You want me to show my slides now? But I'm on – if you look in the agenda, I will talk about that later.

UNIDENTIFIED MALE: Vicky was next.

MICHALE HAUSDING: Yes.

UNIDENTIFIED MALE: Yeah. Jacques, we're trying to [juggle the sequence on this.]

JACQUES LATOUR: Yeah. So next, Vicky Shrestha from Cloudflare.

VICKY SHRESTHA: Morning. Yeah, sorry. We can go to the next slide. It's too loud. Is it? So I work on DNS at Cloudflare and we manage a large global network. We do a lot of authoritative DNS, and we also run the DNSSEC validating resolver 1.1.1.1. Next slide, please.

Basically, this presentation, I'm going to talk about the DNSSEC work that we've been doing so far. We enabled DNSSEC for all customers in 2015. We have been doing [publishing] CDS and CDNSKEY for a long time. We recently also added support for CDS0 and CDNSKEY0.

We see a lot of customers try to enable DNSSEC, but the problem is they don't follow up. So they will enable DNSSEC on the Cloudflare portal but leave it as is, so they don't go through the registrar or registry information and update the DNS, which has been a problem. Next slide, please.

Basically, we see not a very good amount of success rate for the follow-up. The chart shows different TLDs based on the number of [domains] that Cloudflare operates for the customers, and yeah, the percentage of the complete chain of trust by inserting DNS is very low. Next slide, please.

There are multiple problems of why that might be happening. I think a lot of the UI might not be very simple for customers to use or customer might not understand how to do the DS upload properly. It might also be difficult for technically knowledgeable people to follow through.

There were issues where algorithm 13 was not supported by the registrar or registry interfaces, which is sad. Also, there's a problem that custom is afraid their zone would break. Next slide, please.

So what we have now or what we are doing now is basically doing the CDS, CDNSKEY for every DNSSEC-enabled zone on Cloudflare. We believe that this will help boost adoption of DNSSEC by making it easier for customer, so all customer need to do is enable DNSSEC on the Cloudflare portal and we publish the CDS, CDNSKEY and the parent TLD registry, registrar will be able to pull that information and do the DNS insertion correctly. The customer doesn't have to worry about any other detail. So the technical part is all automated. Next slide, please.

So as I said, as soon as customer enable DNSSEC, all of the CDS, CDNSKEY is published at the [edge,] so the parent needs to regularly scan the zone. Also, we can add verification and notification to the customer, maybe have a holding period of like 24 hours depending on what the TLD decides, and make the DS changes. And we also need to

make sure that we do regular scanning so that if the customer does decide to disable DNSSEC, the CDS0 is being picked up quickly. Next slide, please.

This is basically what I just said. Next slide, please. Right now, we have a couple of TLDs, dot-ch, dot-cz, dot-li, dot-cr who support the CDS, CDNSKEY and are automatically [inserting] the DS. We are also working with the registry Gandi, they have been very helpful, and we are obviously talking to more registrars. Next slide, please.

What we are asking is all the DNS operators should start doing this as default. We will also be turning on DNSSEC by default for a customer who uses the Cloudflare registrar, and hopefully, we will open source the software that we use. Next slide, please.

That's it for me. Thank you.

JACQUES LATOUR: Alright, thank you. Next up, we have Ondrej Filip from CZNIC to talk about automated key set management.

ONDREJ FILIP: Yeah. So hello, I'm Ondrej Filip from CZNIC, domain.cz, and I would like to talk about a way how we handle this issue. So next slide, please. Well, and I think Jaromir presented it quite clearly, we have quite good percentage of DNSSEC-signed domains, but we understood that we can get still a little bit more done, and not by going to all individual users but by enabling this feature.

We know that at the time we started, there were about 20,000 domains that were signed, but the DS records were not published or they were not in our database. So it seems that there was probably some problem, and we identified several problems.

Although we really tried to educate all the registrars, their support is sometimes not really optimal. Some of them are perfect and it's great, but some of them really are not at the level that they are helping the users signing domains.

And also, even if all DNS providers somehow support DNSSEC, those domain name holders are not aware of it and they don't know how to transfer key material from DNS provider to the registrar. And also, there was no direct relationship.

And the last thing, some of you probably remember the great time where you just configured your zone, put it into your name service, submitted to a parent zone and that was all and you could forget it. And we want it to return to those great times again, but with DNSSEC and signed zones of course.

So for this dream, we had to change two things. Next slide, please. First was implementation of Knot DNS. So we implemented this feature into Knot DNS, so Knot DNS now if you set your DNS signing as automated, it starts periodically signing your zones, but it also is able to publish either CDS or CDNSKEY record, and then it waits for the parent. So it periodically checks preset nameservers for existence of DS records, and if those DS records appear, it finishes the key rollover.

It can even do algorithm rollover by this way, so it really works very smoothly. You have to either configure all authoritative nameservers or some set of validating resolvers, so it's up to how we want to check that the chain is complete. Next slide, please.

Also, we support just single key signing, it's a little bit simpler, and maybe for some domains, it's enough. Support shared key for more domains if we wish so, but you can also have individual [keys] for all domains. And as I said, it can even do an algorithm rollover. The only thing, if you want to remove the key, so the delegation of kind of empty CDNSKEY must be done manually. This feature was added in version 2.6, and it stays there. Now we have version, I think, 2.7.3 or something like that. Next slide, please.

And the other part of the chain is, of course, the registry. So before we started the implementation, we opened discussion with all registrars and we showed them like three possible options. Either do not implement it at all, implement it on the registrar level or implement it at the registry level. And surprisingly, those guys decided that they will be happy if we implement the things and they don't have to do anything.

So of course, we accepted the role and we said yes. So we will check, and we decided to check CDNS keys in all domains that are in dot-CZ zone, and we will create DS records from that. So we use pull model, and this feature was implemented in a different version, 2.3.2. I think, again, we have a little bit bigger version. Next slide. Yeah, I'm sorry, I'm a little bit – I wasn't synchronized with the slides. Next slide, please.

So that was the implementation in the threat registry. Next slide, please. So, what do we do? We scan daily all domains in our zone, which is about three hours for dot-CZ, and we are trying to find CDNS records.

There are three categories of domains. Domains without the keys record, then we need to have some bootstraps, so we scan the domain for the next seven days, and if there is no chance, then we believe that the CDNSKEY record is probably okay and we create DS record.

Then we have already bootstrapped domains, so if there is any change, we just immediately change to DS records in zone and continue. And also, there are some domains that have DS records already done by registrar like manually. So again, if the information is complete and all the [inaudible] is okay, we can again automatically switch to DS record and delete that record in our database. Next slide.

Here are some statistics. Again, compared to the number of signed domains in dot-CZ, it's really tiny, I think .2%, but it's growing and we hope we will help the other domain name holders to support DNSSEC. So it's something about 1700, again, compared to 700,000 signed domains in dot-CZ. That's tiny, but it's good that we are able to help those folks. Next slide.

Yeah, you know, FRED open source registry system is not just used by us, it's used in many other countries in the world, and good news that one of those countries decided to follow our path and also implement [into] system. So Costa Rica was the third registry that implemented this automated [key set] management. Next slide, please.

So what are next plans? We are just tuning the system. We think we should add more locations for scanning just to make it probably more secure. We are still playing with how to inform the domain name holders about it, because whatever we send them, it's confusing to them. So we should do something with it.

We are also considering implementing the push model, but we wait for RFC to be done, so I don't know when it's going to be, but we can ask somebody. We talk to DNS providers. I think the main input came from Cloudflare, so we hope they will set up DNS [inaudible] referring to. That would be great.

And maybe one important topic which wasn't covered, there was an interesting presentation during the last RIPE meeting. Trust me or not, it was presented by a guy called Ondrej. It's nobody you know. It's Ondej Caletka but it's also from Czech Republic, from the Czech NREN, and he was talking about reverse DNS – [like I think these systems reverse DNS as well,] because DNS can be used for not just translating names to numbers but also in the reverse order. So having such system in RIPE database or in the reverse zones would be nice. So that's also something we could probably maybe talk next time.

And next slide, and this is all from my side. thank you very much.

JACQUES LATOUR: Thank you, Ondrej. Next up is Michale Hausding from SWITCH, talk about automating DNSSEC with CDS.

MICHALE HAUSDING: Okay. Thank you, Jacques, and thank you, Ondrej. We just turned it on two weeks ago, and we're just in the phase where we have the first domain names being put into the dot-ch zone tomorrow. We talked to CZNIC before and we got a lot of helpful impact, and I will most likely talk about what was our motivation and what is our experience so far.

So first of all, DNSSEC adoption in Switzerland is rising, but nevertheless, we think that we can do something to promote it. And we talked to some of the registrars on why they [not] turn on DNSSEC, and basically, there are two things. One thing is they have a registry system who sometimes is not capable of automating DNSSEC or would require a lot of implementation, and the second thing is that they fear the key rollovers.

So if we look at how it worked before, we had a registrant, and he generated a key on his nameserver. The key was then pushed to the registry. For that, the registrant needs to get the key, or the DS, out of the nameserver. He needs to upload it to the registrar, sometimes in a web interface, or sometimes even via e-mail, and the registrar pushes it to the registry via EPP.

Some registrars for example only can do it via e-mail on working days, so in case you have a mistake, yeah, you really need to wait until your registrar is back in the office on Monday. After we activated CDS, it's much simpler, you just need to turn on the signing in your nameserver, which is not the hard part, at least we think that, and the registry can get it from there. So at least we, in theory, can get the key and then publish it. But there are a few decisions to make if you do that.

If we look at the motivation, why we do this, it makes automation a lot easier because the moment you publish the key, the registry can get it and there's no other system involved. The rollover, so if you choose to make rollover, you don't need to acre about it, because yeah, as we heard, Knot does the rollover automatically, and with CDS, we also have a mechanism to update the zone, and apparently for the rollover.

And we also want to show that DNSSEC is not so complicated to use. So if you look at a normal – this is not authoritative server, it's just one line you need to add to the configuration of your zone, and you sign it. And a good thing with Knot is that it has a lot of defaults that even enable the CDS publishing automatically. So for KNOT, you just need to add one file [and] one line, you turn on the DNSSEC signing, and if the registry supports CDS pull, you're done. The same is for PowerDNS. I think there you have two lines of configuration you have to change.

We follow standards. We asked CZNIC what to do, and so far, we didn't have any problems. The question always is how exactly did we implement it, and that's something that's not in the RFC. We have basically two use cases, no, three.

One use case is the zone already validates, then it's pretty easy. We validate or we get the CDS record from the child zone, we validate it via DNSSEC, and if the CDS doesn't change for three days, we update the DS record set in the dot-ch or dot-li zone.

What is more complicated and what caused more time to think was how do we bootstrap. So if a zone is not sign, or let's say if a zone doesn't have DS in the parent zone, then there's no initial trust you can start

with. And so we said, okay, to bootstrap that, we need to build that initial trust, and so we said, okay, we have to have some trust at least in the authoritative nameserver, and we said, okay, let's check all instances of the authoritative nameserver on the IPv4 and IPv6 address. If they give the same answer for the CDS record, that's already something we maybe can trust.

The second thing is to prevent spoofing, we query them only over TCP, so there's no way to spoof the answers via UDP. We also do it for three days, so we think that you might be able to do a BGP hijack, but to BGP hijack all of the nameservers for three days, that's something that most likely will be noticed. And we also do the queries from different vantage points, so it's not just one point from which we query from different points. So it would be harder for an attacker to BGP hijack all of these routes. And if we see a consistent answer for the CDS record set for three days with TCP on all servers from different points, then we put it into the parent zone.

There was a question this morning, what do we do if DNSSEC validation doesn't work? In case we use the pull model, we check, does the zone validate with the DS we got from the CDS record? And if it doesn't, we don't pull the DS in the parent zone. So basically, you cannot break DNSSEC by publishing a wrong CDS record. You still can break it with the push model if you send us a wrong key over EPP.

Yeah, we support only some algorithms and some digests, including a digest zero for deletion. We don't communicate via e-mail, so we talked to CZNIC, and they said usually if you send an e-mail to the registrant,

he will not understand what that message is. If you send a message to the technical contact, most of the cases, he also – he is not technical. And that's why we choose a different approach. We set, "Okay, let's give every domain holder and every technical contact he possibility to see what is the status of the CDS publishing process," and we put up something called the CDS status check that you can type in the domain name, and then you see –

Okay, so you can go to – what's the URL? To nic.ch/cds, and then you will see it. But basically, we will give you all the information. We found a CDS record, it's valid, we queried it one time, we queried it two times, and we give you the date when we put the DS record in the parent zone.

That works for bootstrap, that this image – and that one is working. That's for the rollover. One of the registrars enabled it two days ago, so currently, we are in an opt-in period where the feature is available but registrars must tell us, "Yes, please, turn it on for my domain names." And one of the registrars already did ,and this is domain that is a domain name that is hosted with Cloudflare, and we queried it today three times, and tomorrow, we will put it in the dot-ch zone. So this will be one of the first domain names that we got via the CDS system.

Numbers, it's about the same. We have 2% of the DNSSEC signed domains, so currently, we have about 57,000 DNSSEC signed dot-ch domain names, and we found CDS records for about 900. And if you look at the DNS operators on the right side, we have Cloudflare, we have Google Domains and a smaller hoster who is not a registrar who

published these CDS records. At latest in January, we will put the DS records in dot-ch zone. Thank you.

JACQUES LATOUR: Alright. Thank you, Michael. So next up is Richard Merdinger from GoDaddy, and then he's going to tell us all about how GoDaddy fully supports – no, okay.

RICHARD MERDINGER: Well, that's awesome. First of all, I'd like to start by thanking Steve Crocker for inviting me and bringing me into the discussion as a registrar. GoDaddy obviously is a very large registrar, but by proxy, that means we're also a very large DNS service provider.

We do have a slide that will come up eventually, I hope, but when you think about GoDaddy, you think of us as a registrar first and not the DNS provider. We've had full DNSSEC enablement within our DNS solution for many years now. I worked with Steve a long time ago to get that going along with PIR.

And we are at the point now because we are a registrar with direct connection to the registry, for the DNS zones that we manage, we have the one-click turn on DNSSEC, automated key rollovers, things of that nature that work within our ecosystem.

I think rightfully so, we've been relatively focused on our ecosystem, and it's about time we start focusing on the broader ecosystem, because as I mentioned, GoDaddy is really – we do many services, but

we do a service that is registrar and we do a service that is DNS. Cloudflare also is a registrar and has DNS.

One of the things just in general, we support the idea of the CDNSKEY and the automated updating of DS records directly from the DNS service provider to the registry. However, we do covet the relationship between us and the registry when it comes to changing the paradigm of the services we're providing for the registrant or for the domain holder.

So when it comes to changing, whether turning on or turning off DNSSEC and instantiating the DS record, we strongly believe that that should go through the registrar to get that in place where subsequent updates happen on an ongoing basis.

And as we all know, with CDNS, this can happen very fluidly, but it's going to require an implementation at the registry level, and depending on the zone size, there may be pragmatic issues [inaudible] scanning zones may take a very long time, infrastructure cost, things of that nature.

So we don't think that there should be just one solution to this. We think that there should be a set of solutions. Some of them can be [very –] for those that choose to make the investment in the implementation, that this solution is just fine.

Our chief architect, who had hoped to be here today but had t o leave yesterday, developed internally and then made more public a protocol referred to as Domain Connect. Domain Connect is a paradigm by which

to enable service providers such as hosting providers, any type of provider of a service can configure via templates DNS through the DNS service provider.

So for example if someone went to Wix or Weebly or what have you and instantiates a website, they're able to put in their domain name, the Domain Connect system determines out of a special configuration [in] DNS where the provisioning provider is for that, where the DNS I supported, and there's a system in place that'll allow the Cloudflare, the service provider, to automatically apply a template to the zone that they don't own over at the DNS provider. So it's a way that decouples the service from the DNS provider and kind of tries to remove the middle man and really make it a seamless implementation.

We have talked about augmenting the Domain Connect product – it's not a product, excuse me, the Domain Connect service such that DS records could also be updated such that the registrar would be considered – and separating GoDaddy's DNS from this, the registrar service would be subscribed to Domain Connect, the DNS provider would be subscribed to it, and the DNS provider could initiate a DS record update to this registrar who would then propagate it via EPP to the registry, the idea being that the registry in question here did not opt to implement this other mechanism.

So in general, we're in favor of doing what has been presented. We do want to be very careful about allowing paradigm changes to the DNS instantiation for a customer. But once the decision has been made to operate under DNSSEC, we don't want to be in the way and preventing

the ongoing update that should be taking place. That's really all I have, so I don't have a lot of slides [inaudible].

JACQUES LATOUR: Alright. Thanks, Richard. Alright, I only have 100 questions, so not too sure what to say. I guess the first one – I'll start with my question and then we'll go to questions, but for Vicky, you said Cloudflare is publishing a new record type, CDS0 and CDNSKEY0 as opposed to having an [old] CDS and CDNSKEY.

VICKY SHRESTHA: No, Cloudflare is publishing the CDS and CDNSKEY for all DNSSEC-enabled zone. We recently added support for CDS0 and CDNSKEY0 if the customer decide to disable DNSSEC on the [portal.]

JACQUES LATOUR: So it's not a new record type, right?

VICKY SHRESTHA: It's just something that's in RFC – what's the number? 8078, to disable DNSSEC if they want to.

JACQUES LATOUR: Alright, so any questions? Peter.

PETER KOCH: Hello. My name is Peter Koch, I'm DENIC's chief skeptic. So when looking at the motivation for supporting CDN, CDNSKEY, I heard Michale say, I think, one of the reasons was that rollover is so complicated. Then why do it? If I remember correctly, one of the motivations for the whole key rollover was to train staff that in case of an emergency key rollover, people are enabled to do the rollover.

Now what we've done is we found out that going to the gym is a bit cumbersome, so we've hired students, a.k.a. [inaudible] automation, to do this going to the gym for us, and we are expanding on that automation. It sounds a bit odd to me. What remains is the emergency key rollover, how would that interact with the method. And we've also had lots of work on DNS operator transfer which still would need interaction with the registry and registrar. How would that be addressed with this CDS or CDNSKEY model?

MICHALE HAUSDING: Okay, more than one question. The first question is the key rollover. So first of all, I think the key rollover for a second-level domain is quite different than the key rollover for the rootzone or even for the – so I think the key rollover for a second-level domain is different, and so most users, they can turn it on, but hey will never understand how it works. So training there is something you really need to leave to the DNS server. And Knot does the key rollover if you want to do a key rollover, but I think there's no training on the registrant side for a key rollover compared to the TLD key rollover or the root rollover.

The second question was emergency key rollover and DNS operator change. That's something – in case of an emergency, there are two possibilities. One is you do the emergency key rollover. With CDS, you have to do that on your DNS server. So as soon as you roll over the key in the DNS server, CDS picks it up, and yes, there might be some delay.

The other possibility is to go unsigned in the case of an emergency even if we don't recommend this, and that's why we also implemented the algorithm zero with which you can indicate to the parent zone that you want to go unsigned. So that's also a possibility, to do it via CDS. We only scan the zone once a day, so that will give you a delay for up to one day until we realize that you want to do an emergency key rollover or you want to go unsigned. But you always have the possibility for dot-ch to push a new key or to remove it from the zone, but that only gives you some time, so you still need to do it.

JACQUES LATOUR: Steve, you had a question?

STEVE CROCKER: Yes. Thank you, Jacques. Three things. I want to respond directly to Peter Koch's provocative question, and thank you, Peter, for it. My understanding of rollover is that it's a good practice and ought to be done on a regular basis, not only for emergencies, that algorithms and keys both have natural lifetimes, and even though one could argue that you don't need to roll the key because it's good for a very long period of time, that's a bad way to develop, to run a long-term sustainable

system. So I take issue with the point that key rollovers is only or primarily for emergency, and rather that it ought to be done on a regular basis and normally.

Second thing is with respect to changing DNS operators, I think that's an important capability to have. Recently, I was exposed to a presentation by [Schuman Hawk] which covered not only change from one operator to another but having multiple independent operators serving the same customer. And that's sort of a slow version, if you will, of changing from one operator to another, you sort of bring the second operator up and you don't bother to take the first operator out of service. I think that would be an excellent and complete solution to the problem of changing operators.

Now, let me get back to the question that I opened up with. What else is needed? There's a lot of presentations here that are focused on the CDS and CDNSKEY solutions and then having the registries pull the records. The presentations have all [inaudible] been from ccTLDs, not from gTLDs. What's your estimate of what it will take to bring that kind of solution into the ICANN contracted parties space?

And then you have the complimentary approach that Rich has pointed to of providing a push model with appropriate interfaces. And I'll just repeat what I said at the outset, which is I don't have a stake in this except to have the whole thing work one way or another. What I'm concerned about is having it sort of stall, then half of the parties say, "Well, we're doing it this way" and the other half of the parties say "Well,

that's not what we want to do" and then I don't want to see it just get stuck in the middle there. Thank you.

JACQUES LATOUR: Alright. Thanks, Steve. So there's a lot of issues that we need to address, but I think one of the biggest ones is the bootstrap process, how do we initially get the DS in the zone. And that seems to – when we started a discussion around this, people were generally reluctant to grab a CDS and create a DS. And now we started to do that with ccTLDs. In the gTLD, we need –the policy definition around this are very weak, they're very broad. So, do we need to create better language that defines how gTLDs accept DS from registrars, from the child? What's the process to get a DS in the registry? What are the options that we accept or not accept?

And if we know what the switches are, then we can build protocols around them. Because today, we don't know what they are exactly, and it's difficult to build process around this. So I think like GoDaddy – here's an example. If you create a dot-CA domain and the child, the domain is signed right away, there's a CDS record, I know it's a new domain because I just created my registry. If I poll it and I grab the CDS and add the DS right away, that goes against your gut reaction. So how do you feel about – I'm not sure what the – it's the second time we do a policy panel-ish on this, and we're moving forward, but not very fast.

RICHARD MERDINGER: Sure. The context that I just understood was that the registry is part of the [create,] is essentially initiating the instantiation of DNSSEC. Is that correct? So, the registrar, [does it create the registry?] And the registry is part of that multifaceted transaction, [is] instantiating DNSSEC. It still fits with the paradigm that given that that is the policy that the registrar's initiating the creation. So that isn't a problem.

JACQUES LATOUR: But you created a domain, right?

RICHARD MERDINGER: Correct.

JACQUES LATOUR: But you'll always create a domain, right?

RICHARD MERDINGER: Yeah, but if we create a domain and the registry policy is not to instantiate DNSSEC, then a subsequent instantiation of DNSSEC should have to be initiated by the registrar. We brought this together into a multifaceted transaction by saying create leads to DNSSEC. That is initiated by the registrar. The other scenario is that there are two transactions, one is a create and then the other is the instantiation of DNSSEC.

JACQUES LATOUR: Okay.

RICHARD MERDINGER: If I'm being vague, I apologize. It's day 35 of ICANN.

JACQUES LATOUR: No. Okay, so I guess I'll just think about this. So if you create a domain in dot-com and dot-com – oh, sorry.

PAUL WOUTERS: No, go finish.

JACQUES LATOUR: I just want to finish. So if you create a domain in dot-com and dot-com doesn't scan, then GoDaddy could grab the DS and add it automatically.

RICHARD MERDINGER: Yes.

JACQUES LATOUR: That would be great.

RICHARD MERDINGER: Again, we do that for our own DNSSEC-enabled zones today.

JACQUES LATOUR: But not –

RICHARD MERDINGER: But not for other providers. But that's part of becoming part of the ecosystem and [broadening] it beyond our own border that I talked about.

JACQUES LATOUR: So we need to define that.

RICHARD MERDINGER: Yeah.

JACQUES LATOUR: Okay. Paul.

PAUL WOUTERS: Hi. I'm one of the two authors of this RFC for CDS and CDNSKEY. The reason we wrote this was that there was a strong demand for DNS operators to be able to do this without the registries being involved, because often, a customer will change DNS hoster, but they will not actually change registrar at that point. And usually, the domain holder is not aware of any of these technical details, they just change their DNS provider. They don't understand or even know what DNSSEC is. And the whole point was that if you'd move to a DNS hoster that has this capability to do DNSSEC, they won't need to bother the domain holder with it. These are just technical details that the domain holder really doesn't care about, but the DNS hoster who wants to deploy DNSSEC

on default does care about this and wants to deploy this for the client. So that was one of the main use cases of getting this automated completely.

JACQUES LATOUR: Thanks.

RICHARD MERDINGER: If I could respond to that, there are certain elements of – the lines are very blurry on this, but there are certain elements of where the SRS and the registration database at the registry contains attributes of a domain name. Some of those attributes also live, in my view, inside of the rootzone file, the nameservers, etc. So I'm viewing that their nameservers, in other words the selection of the DNS service provider, and the DS records, are really more domain configuration components than they are – even though they live as part of the ecosystem and are essential in the delivery of DNS. [The feedback cut my mind off,] so I'll just stop now.

JACQUES LATOUR: Steve had a question. Final [talk.]

STEVE CROCKER: Thank you. I want to ask you a detailed question which is aimed at trying to bring out whether or not there's some inherent or [inaudible] here. And let me address it to Rich, but it's really for the benefit of everybody. If the registry does the scanning and uploads new

information, that pathway does not go through the registrar. One of the comments that I hear repeatedly from a policy perspective is that he registrar needs to have complete information about the zone and that's one of the reasons why there shouldn't be a pathway that goes around the registrar. I'm sorry, the registrar needs to have all of the information, that's why there shouldn't be a pathway around it.

You're in an excellent position, Rich, to speak to that particular detail, sort of how do you feel if new DS information goes from the child up to the registry and has not gone through from the customer to you?

RICHARD MERDINGER: Thanks, Steve. Sticking with the original paradigm I was saying, it's as simple as the registry scans, they update the DS record, a pull message is created in the note, and the registrant is notified of the change and we could update our systems to be in sync.

There are ways that we can make sure that configuration changes to the DNS – we keep all the systems in sync. I don't know what's going on, sorry.

JACQUES LATOUR: We have – good luck.

RICK WILHELM: Rick Wilhelm, VeriSign. Just a brief point and earlier thing we were talking about about the registration flow. Typically, during the create process, it's, in my experience as a prior registrar and a current registry

that DNSSEC is rarely set up on the initial create. Most of the time, a prospective registrant does not want that to get in the way of the registration path, and at least when I was a registrar, I didn't want that in my cart flow to flow someone down from conversation.

I'm sorry to say I don't know what Rich's cart looks like currently, if you even offer that during the initial registration flow, but most of the time, the create and adding DNSSEC is more likely done in an update process post-create. Thank you.

RICHARD MERDINGER: Very briefly, I'll just say that you're correct. Frictionless purchase path, etc. Many domains are purchased not even to be instantiated. Putting the overhead of defaulting to DNSSEC [does] make sense at huge scale, etc.

JACQUES LATOUR: Is it a one or a two? One is a new item, two is an answer to the existing –

UNIDENTIFIED MALE: So there is an EPP extension where the registry can actually update the registrar with new information, so they can actually relay and say, "Oh, I got this new information from the DNS just so that you can update your records that you have of your client."

JACQUES LATOUR: Yeah. Patrick?

PATRICK FALLSTRÖM: I think we have to separate the initial turning on and off DNSSEC from the actual – which is like depending on whether there are any DS records or not and the update of DS records. And it's a big question of who actually has the responsibility for the zone to work. And that is a very strict contractual agreement between the registrar and the domain name holder. And having sort of things happening regarding turning on and off DNSSEC, that operation, as I have said in the IETF also many times, as people know, that is something that my personal feel is absolutely the registrar's responsibility.

On the other hand, I think updates of DS records when DNSSEC is sort of up and going, that's a completely different story. So I support what I see on this slide if I don't misunderstand them.

JACQUES LATOUR: [inaudible].

RICHARD MERDINGER: I don't know if it's a one or a two, I missed that part, but to that, Patrick, the enablement of the DNSSEC – excuse me, guys, I'm going to skip this.

JACQUES LATOUR: Alright. Any other questions? I think we could go on [for days,] so I want to thank the panelists for the session. Thank you.

RUSS MUNDY: Okay, Jacques, thanks very much for that panel and all the panelists, we really appreciate that. Our next session here, our next panel is on everybody's perhaps favorite subject, especially now that it went well, the KSK rollover. And we have Wes Hardaker and Matt Larson. And Wes is first, so soon as you are ready to start speaking, Wes.

WES HARDAKER: Yeah. Go ahead and hit it. I know we're running late, so that's okay. So I'm Wes Hardaker from the University of Southern California, the Information Sciences Institute. I'm going to talk a little bit on sort of how I saw the key roll go from my perspective as running one of the set of root instances.

So this is really where my perspective shifted from concern in the beginning to sort of calm. Next, please. I have the clicker? Excellent. So this is the overview. I'm going to sort of start with my paranoia and then deciding to face my fears and then seeing what reality looks like.

My initial paranoia was grounded in what happens if the world sort of falls off the net. This could go bad, and so I got sort paranoid and went looking into stuff to see what I can figure out and what happens when resolvers go crazy, what happens when those resolvers that went crazy start sending millions of requests.

And it happened before, so this old graph was actually one of my biggest fears where a previous key roll caused a huge spike in traffic,

and as a root operator, one of the things I worry about was this happening again. So this was where my paranoia all started from.

And that really lead me to analyzing lots and lots of data. Lots of DITL data, lots of interesting things that I could look at. Previously, I talked about finding some misconfigured VPN software, and hey fixed their problem so that their users didn't have a problem during the key roll. I need to write to them and see if they heard any complaints from users that failed to update their software, but I haven't done that yet.

Today, I'm going to talk about sort of my failure to find much else. So that was really good news, and it 's sort of one of the reasons that the key roll went so well. So to find out who's in trouble, like who might actually be impacted, that's one of the things that everyone has struggled with. Jeff Huston has done a lot of analysis in that area, I have, and a few others have too.

But you really need to know two things. You have to know what keys a validator was actually configured with, and there were a couple of sources of information from that, one of which was RFC 8145 and the other one was sort of the newer KSK signaling. Neither of those are complete signals, and the reality is it's impossible to get a perfect dataset, and this is sort of one of the failures of – I think if we were going to redesign things today, we'd change how we look up keys to make sure that we had that information from the receiving side.

The other thing that you need to know is, are they validating? And there's a lot of debate about whether it's possible to measure whether a resolver is validating just by looking at their traffic, so I thought, well,

let's try. So a little bit of background – wow, that font's really small – about how resolution works with respect to a validating resolver. So first off, a resolver will send to a root, "Hey, do you have www.example.com?" If you've ever been to the DNSSEC for Beginners, you've seen this in the skit, right?

And the root actually responds back with the nameserver records for com, the glue records for com, and it actually includes the DS record and the signature on all of this data. Well, not the glue.

Interestingly enough, frequently, the resolver immediately turns around and sends back a request for the DS record back to the root for com. It doesn't need to do this. It actually already has the signed record that was given to it, but it's sort of unclear which resolvers do this and which ones don't. But I talked to Mark Andrews from ISC and he confirmed that BIND, for example, still does this. They haven't gotten to the point of checking the data in the additional section to see if they can keep it.

So this lead me to the hypothesis that a validating resolver that is querying for legitimate data within the root, 50% of the requests would be for DS records. The other 50% would be for their original question. Now, the next thing they're going to come back with is the DS record. They really shouldn't be querying for anything else. There's nothing else in the root that they should be querying for.

So the first step is you have to remove all of the stuff that they're querying for junk records. So this is an interesting graph. The purple line in this graph – this was done by one of my summer interns that analyzed

DITL data to come up with a purple line is all of the multi-label stuff. So looking for www.example.com or whatever, or [inaudible] doesn't exist.

The orange line is the junk data that Chrome generates trying to determine if it's behind a paywall. And so I find it fascinating that one of the biggest signals to the root is actually Chrome junk data, because every time Chrome starts up, it fires off three random garbage strings to the root. The other three lines below it are all the other single-label queries, so like com, net, org, university, whatever.

So this is basically the results. So if my hypothesis is true, we would expect that if you go looking for all of the data that an address sends to the root servers, that 50% of the queries would be DS queries. So the horizontal axis here is basically the percentage, and the vertical axis is the number of hosts that sent that percentage.

So you see two basic, gigantic humps. One is right at 50%, which is what my hypothesis was about, right? And so as hosts send more and more data, that gets more and more closer to 50%, because remember that this is from the perspective of one root's data. So on September 25th, so right before the key roll, and the stuff on the left, I'm not sure about. I'm not sure why people would be sending 10%.

But assuming that my hypothesis is correct, I decided that, well, let's go look at basically that bump. Let's go look at all the stuff in that bump. So let's narrow basically down what we're going to look into, down to things that sent between 25 and 80% of the time was DS records. Assuming that these are probably validating resolvers. So these are the ones that I want to check. And I looked at it again on September 25th,

so this was right before the key roll. Again, my paranoia was still flying high at this point.

So the next question is, which of those validators in that gigantic bump were sending only KSK 2010 8145 validation signals of the keys that they were using? Those were going to be the ones with problems. This is basically what I did of the previous VPN software that I looked at too.

There was 45,806 matching resolvers in that bump. The total number sending the KSK 2016 key, in other words support for the new key, was 420. Unfortunately, that's a whole bunch that did not fall in with the new key. Anyone want to take a guess at how many were sending the old key signal? 12, which is down from 18 from the month before.

So only 12 validating resolvers that – determining my 50% rough guess, only 12 validating resolvers were actually sending the old key. Now, granted, there was only 420 new ones, so clearly, there was a lot of missing signals too, but you can account for anything. But that discrepancy between 12 and 420 suddenly made me feel a whole lot better, and Matt's team going forward and doing the roll a week later, I was definitely much better. So my naysaying before of, "Oh, no, the sky may fall" was definitely wrong. So that happens. I get it wrong all the time.

So my next question is sort of what DS records were they sending, what were they querying for? Reasonable resolvers really should only send a DS query for com and should only send a DS query for a TLD, never for something like example.com. Are resolvers really reasonable? No, they're not. So actually, when I went and dived into those 12 resolvers,

a lot of them were just junk queries. So there's a lot more random analysis to be done.

So what did real life look like? I'm going to show you actual graphs from our operational network. You don't have to understand what the lines are, but basically, these are all graphs showing traffic to USC ISI's root server. This is just before the roll. If you look really carefully on the right-hand side, you'll see all the lines right at the time of the roll climbing upward. And that was a scary few moments when everybody went, "Uh oh, the sky may actually fall."

But this is the bump a couple hours later, and you can see it was a very short bump, and then everything went back to normal and we got an absolute flat line after that. In this case, a flat line means that thing lived, not died like in the medical world.

So, in conclusion, the world seems to be like they're finally getting better at two important things, right? They're finally getting better at updating software and updating configuration, both of which were kind of required for this event to go smoothly. And I do believe that the extra year that we took was an incredibly wise decision by, again, Matt and his team, because I'm not sure it would have gone quite so cleanly a year ago. But that gave enough time for a lot of things to change.

And then the last thing is that resolvers are weird. We think that we understand them, but anytime you go diving down to look and see what they're actually doing, you don't. So the RSSAC caucus right now is embarking on a study to look at how resolvers behave, and I'm really looking forward to that data, because the reality – strange things are

afoot in the circle K, and if anybody's seen the Bill and Ted's Excellent Adventure, that's a quote from. But it's strange things are afoot in resolvers, and we really need to understand them better to make sure that all future things that we want to do in the DNS world go smoothly. So thank you.

RUSS MUNDY: Okay. Thanks very much, Wes. Let's move on to Matt and take questions at the end.

MATT LARSON: Hi, everyone. I'm Matt Larson, VP of research at ICANN in the office of the CTO, and I was involved in the project to roll the rootzone KSK. Well, I'll tell you what, in case you're in suspense, we did roll the KSK. It happened on time, as planned, and I have just a little bit of data for you to talk about it. It's loading.

RUSS MUNDY: Aren't computers wonderful?

WES HARDAKER: While we're waiting, I'll throw out one more fact for you, Matt. I went and looked just two days ago at data to see how many resolvers fell into that same 50% bucket. Ten. So we're down two. But they're not the same ten.

MATT LARSON: Okay. Adobe is taunting us. There we go. Okay.

RUSS MUNDY: So, it's up in the room but hasn't quite reached the Adobe Connect room yet. But I think we'll ask Matt to go ahead and let the room catch up.

MATT LARSON: Okay. Alright, so as I said, the rollover happened, and it happened with the publication of a rootzone on October 11th at 16:00 UTC. Here's the timeline of events, just if that's at all interesting. We had a three-hour window ahead of time where we spent more time than usual looking at the rootzone. Usually, ICANN, the IANA function doesn't inspect the rootzone at all before it's published, but in this case, VeriSign looked extra carefully at it, ICANN looked at it. We just really wanted to make sure. We had a go/no go call within ICANN to confirm everybody was ready, and then it happened on time.

So we did it from Amsterdam, that's the VeriSign and ICANN folks up there, and we all wanted to be at DNS OARC and we didn't want to be on a plane flying from the KSK roll to Amsterdam, so we said, well, let's just do this thing from Amsterdam. And NLnet Labs very kindly gave us space. They were very gracious hosts. So thank you to NLnet Labs.

And then that young guy in the lower left is holding a copy of the rootzone when it was first signed in 2010, and so there was insistence that we had a dramatic reenactment on the lower right. So for what it's worth.

One thing that we did see is if you look at the number of DNS key queries to the root, it did increase. I have an animation, and unfortunately, I can't show you, but I have some screen captures from that. So I apologize if these slides are illegible. What you want to look at is the upper left.

These are DNS key queries from each individual root server that reports stats, data to ICANN, and the upper left is the aggregated. So that's all roots that we have data for in the upper left, and what you can't see is that the scale changes, unfortunately. So it's going to look like things stayed the same, but they don't. I can't even see here. So that scale tops out at about 1400 queries per second, and on the far right of the graph is the KSK roll, that's when the KSK roll happens on October 11th.

So here's then leading up to the 48 hours after the KSK roll, and now where the red arrow shows the top of the graph of the scale is now about 2500 queries per second. And then if we go to right now, it tops off at about 4000 queries per second. And the other thing to know about these graphs – let me go back – the yellow line that you can barely see is a week ago. So if you're wondering, well, is this just some sort of curve that's continuing forever, the answer is no, because as of now, when it's been more than one week since the rollover, the yellow and the red are the same. So we've reached some sort of plateau.

So we've tried to figure out what exactly is going on, and we've just started this research, but Roy Arends on the research team has done some great work on it. And we certainly expected more DN key queries because based on resolver behavior that we've studied in the lab, when

a resolver has as stale trust anchor, it tries desperately to get someone to tell it a DNS key that matches what it has. And of course, it's never going to find that, so it just queries and queries. And the amount that it queries depends on that particular kind of resolver.

So what we decided to do was let's look at DNS key query behavior before and after the roll. So October 10th and October 14th. So there were just over a million, almost 1.1 million unique resolvers asking for a DNS key over the entire four days, but we see 155,000 of them that asked both on the 10th and on the 14th. And then 85,000 resolvers sent a request at least once a day. And this is looking only at L root, of course, because that's what we have the actual packet by packet data for.

So let's take a look at these 155,000 resolvers that talked to us at least on the 10th and the 14th. So this graph takes a little bit of explanation. On the lower – well, here, let me first give you explanation. So the X axis on the bottom is how many queries a resolver sent, how many DNS key queries it sent on October 10th, and the Y axis is how many it sent on October 14th.

So each one of these dots represents a given resolver. And what we would want in the green area would indicate that it sent the same order of magnitude queries on the 10th as it did on the 14h. Right? X and Y are the same. And this is log scale, so every one of these squares is in the power of ten. So you can see that the concentration in the lower left is resolvers that send a relatively small number of DNS key queries. Let's see, that's two orders of magnitude, so, you know, from 10 to 100, and

then you see as we go up to the right, we have more and more high volume.

So what is in that purple-ish circle? Those are the ones that sent significantly more on the 14th than they did on the 10th. And as you can see, it's potentially an order of magnitude. Well, it is an order of magnitude more based on how it's clustered, right? It's three orders of magnitude on the 10th and four on the 14th. So those are the ones, that's the 155,000 set that we're looking into.

And this just shows how many resolvers fall into which bucket in terms of magnitude or change, but let me go straight to the slide. So the upshot is the final three bullets – so this is the summary of what I'm getting to here, is that almost 96% of the resolvers stayed within an order of magnitude up or down in terms of the number of key queries that they issued.

But there is this little chunk of 2000 of them that sent more than an order of magnitude, and some of them went bonkers. I mean they're continuing to send a lot of DNS key queries, a significant increase. And then it's also worth noting that an even larger number , 4500, are sending fewer DNS key queries. So this is just the early phases of this investigation, so I can't tell you any more than that. We're going to continue to look at this just to try to understand. Hopefully, this won't be one of those mysteries like the traffic that old root server addresses get that never seems to go away, like background radiation. So maybe we'll now have background radiation DNS key queries as well on the Internet to deal with forever.

That didn't come through. That's the graph – oh, it's kind of there. That's alright. Well, it doesn't matter. I don't like the data anyway. So this is the RFC 8145 reports. That's the dataset that ultimately lead to our decision to postpone a year. And what this just shows is that the more of this data we get, the less we really wonder what it's telling us. There's actually a little bump for the KSK roll indicting that there were now a higher percentage of resolvers that had only KSK 2010, but then that bump has gone away as of now, so I showed it just to show it, that we're still watching, that you can still go to this website. For the foreseeable future, we're going to continue to publish that data, but I do think it's of questionable value at this point except as a curiosity to research.

So there were, as everyone probably knows, there really was very little impact that we could tell as a result of the KSK roll. There was only one very minor trouble report that somebody sent to ICANN. There are a handful of issues that we heard about via Twitter and mailing lists, and it was on the order of one person said, "Oh, yeah, one of my monitoring things that I do broke and then I remembered the KSK roll and I fixed it." So it's really minor stuff like that.

There are two outages that may be the result of the KSK rollover. The timing of both of these is suspicious, and the Irish one in particular, the first one, they do mention DNS in their public report of the outage. So very suspicious that the Irish one is KSK-related, but no one is talking. So I'm hoping if I give this presentation over and over again, maybe they finally will decide that they should poke their head up and say yes or no. I'd just be interested in a yes or no. Don't have to get the details, just

be curious to know if the KSK roll was the issue or not. And then there's this ISP in Vermont, that, again, suspicious timing. I've reached out to them, and they've said they're potentially going to get back to me. So we'll see.

If Ed Lewis on my team were here, he would remind us that the KSK rollover is not over. There's more to go. At the Q4 root KSK ceremony, we'll generate the signatures that will revoke the KSK 2010, and that happens on January 11th. That is the day that we publish the rootzone with the revoke bit set in KSK 2010, and then an extra signature as part of RFC 5011, if the key's going to be revoked, it has to sign itself to show proof of possession of the private key.

So this means we are going to have a slightly larger DNS key response from the root because there'll be one more RRSIG, and this will be a historical maximum for the DNS key response. Now, we really don't have any indication that the larger DNS key responses that we've had or some time now is part of this – well, actually a year and a half. It was July 11th, 2017 that we published four DNS keys, two KSKs, two ZSKs in the rootzone for the first time, and there's been no reports of any issues for that. So I think a tiny little additional bump is probably not going to be a problem. I'm not worried about it, but it is going to happen.

And then on March 22nd, that's when we pull KSK 2010 from the root zone, so at that point, we'll have only one KSK. We'll be back to that steady state of one KSK, and then just two ZSKs at the quarter boundaries when we do the ZSK rollover. And still, we're not done at that point, because the key material for KSK 2010 remains on the

hardware security modules in the two key management facilities that ICANN runs, so in Q3 and Q4, we will delete that key material.

So that's my last slide. I don't have much to say about forward looking, except that now of course that we're almost done with KSK rollover, it's time to begin that discussion, and I'll just reiterate what I've said all along, that ICANN, we intend to work with the community on the KSK rollover future directions, because it's not up to us to unilaterally decide, it's up to us to work with the community.

I do think what will probably work out best is if we present a strawman just to get the discussion going. I think otherwise, we'll just be looking at endless lines at microphones forever with people suggesting how they would do it. But I think until we have something concrete – and it's not just – there's several things that we need to talk about.

There's frequency of KSK rollover, there's issues like do we want a standby key, then there's also the algorithm roll, and these are all issues that are related but separate. So we need to figure out how to frame that and present it to the community so that we can have a productive discussion. So I think you can look for that soon. I would hope by Kobe, we could be talking about that, or at least beginning that discussion. And that is my presentation.

RUSS MUNDY: Thank you, Matt. Thank you, Wes. We appreciate those very much. And we are right up against our time. And just to remind folks, I apologize, we don't have time for questions, but we do have to clear the room

because this lunch is not our lunch. Our lunch is out the door, up the escalators two levels, and it's the banquet hall. And don't forget your lunch ticket. But please do take all your equipment and everything, and then rejoin [inaudible].

[END OF TRANSCRIPTION]