HYDERABAD – Tech Day (Part 1)
Saturday, November 05, 2016 – 11:00 to 12:45 IST
ICANN57 | Hyderabad, India

EBERHARD LISSE: Good morning everybody. For all you don't know me, my name is Eberhard Lisse. I'm the chair of the ccNSO technical working group, and I run dot NA, which is the ccTLD in [inaudible]. And for the ones that don't know me, I have a day job. I'm a gynecologist by profession, but fortunately I don't have to do my night job anymore.

I don't deliver babies anymore. I chair this technical day, as usual, and what we do in the first five to 10 minutes is quickly go through the agenda. This time, it's going to be slightly different from our usual methods. We usually had two sessions of equal length, interrupted by one hour lunch break, but because ICANN has changed the format of the meeting on a global scale, we sort of adhere to this.

We have, first five presentation, four presentations, and then we have this one and a half hour lunch, and then we also have a coffee break at 3:00, which we usually wouldn't have. This is the coffee break that coincides with all of the others, so you can go and mingle and meet with colleagues.

And we felt, we adhere by it. If we had not one of our presenters beg out on the last minute, we would have, of course, used the coffee break as well. Let's go to what we're doing first. Roberto [inaudible], Roberto is going, from Google, he's going to speak about the new open source registry platform.

And as soon as I mention it, everything fails. No, no, leave the agenda on please. Leave the agenda on please. I want to go through the agenda.

We have, as you all know, we have two other open source registry platforms, Crocker and Fred. We have given both of them enormous bandwidth here to present and to follow-up. So, I feel it's very important that we give the same amount of time or opportunity to Google, and since it's an open source platform, we have even more justification to do this.

After that, Michele [inaudible] and his colleague from [inaudible], are going to speak about a little gadget they have developed, which I personally have no use for, but I find very cool, and it's going to be quite helpful to IDN domain registrants.

The Christian [inaudible] from dot NL is going to speak about a non-validating resolver, validating resolver that they have got developed. And then we have the host presentation. We have two presenters. I haven't met the two presenters personally. If they are in the room, can they please raise their hands?

Not yet, so I will email then. Then Ray [inaudible] is going to speak about some stuff he has developed. He always does cool stuff, so whenever he says he has got something for us, I'll put him on because he's always very interesting. Diego [inaudible] is going to dissect the source code of [inaudible], which has been published.

And since this is responsible for the huge DDOS attack that we saw a few weeks ago, I must think it's important that everybody who understands C has a bit of a look of what's going on there. Then, [inaudible] is going to speak about DNSSEC automation of [Fred], one of the other open source registry platforms that I invite you.

Then Dave Conrad, or one of his staff, is going to present about the recent planned root zone KSK rollover. Jacques [inaudible] is going to speak about a protocol that they're working on, and that they are pushing forward, which I really don't understand, but that's what we have Jacques here for, to explain.

Then Joe Waldron from VeriSign is going to present about the registry verification framework. They proposed this, I think, one meeting back, but couldn't get some license approved in time. So, we now are giving them the opportunity to do. And then, the technical contact from dot KM, which is the commerce, emailed us and suggested that they give a presentation.

Are you in the room? If so, raise the hand. No. Dot KM, as you know, is a very small ccTLD, and I'm always very keen on giving a forum to the smaller TLDs, because they have got the same, or bigger, problems then the bigger ones, but less resources to deal with them. So, it's always good to hear from them.

And that means, without much further ado, and about five minutes early, I give the floor to [Richard] Roberto from Google, to talk about their open source registry. We are not strapped for time. I would actually prefer if everybody uses up, more or less, all of their allotted time, and not finish 20 minutes early or something, but we have a large lunchtime, so if you run a little bit late, it's not going to be a problem.

RICHARD ROBERTO: Good morning. My name is Richard Roberto. I am on the Google registry team, and I'm here to talk to you today about Nomulus, which is the source code for a software that Google uses to run its registry business. And we open sourced it about a month ago. We'll talk a little bit about our motivation for that, talk about some of the features.

We'll do a little bit of a demo by way of a screen cast video, and then we'll take some questions. So, what is Nomulus? It's a sort of green field registry system that we decided to build from scratch when we decided to get into the registry business.

We built it initially with the intention of open sourcing it, eventually. It runs on Google's app engine cloud product. It uses data store, which is a highly performing no SQL database. It supports all of our TLDs as a full range of like gTLD required TLD functionality. Obviously, it supports EPP, WHOIS, we actually have an earlier version of RDAP, that's sort of following the RDAP track.

It has some reporting capabilities. It has got trademark protection built into it, and we have data escrow support with automated crypto signing and uploads. It's written in Java, and it is written, released under the Apache 2.0 license.

And why open source is one of the questions, I think, has been asked numerous times. And you know, when we first got started on our project, maybe three or four years ago, whenever that was, and we decided that we were going to build our own registry from scratch, we looked around for open source implementations to get ideas about how people have interpreted or [inaudible] because [inaudible] is not always unambiguous, and sometimes things are underspecified on purpose, and there is a lot of room for, you know, business decisions to be made about how to implement policies and that sort of thing.

So we looked around, and we found a couple of options that a doctor had mentioned, and we found that some of these unanswered questions weren't really clear in terms of what may have been sort of industry standard, or what registrars may have preferred and that sort of thing.

So we took a lot of guessing, and we wanted to open source our version of this, so that there was yet another example of how someone has sort of dealt with these ambiguities. And we thought that even if people don't adopt Nomulus as their software stack, they could at least see how we dealt with some of these corner cases, and these sort of underspecified parts of these specs, so that they can decide whether or not that's so they would want to do it.

And therefore, make it a little bit easier for registrars to integrate with the system, because of how similar things would be. We had some other platform goals, we wanted to make sure that we had an opportunity to demonstrate some of the advance features of Google's cloud. We wanted to make sure we had what was sort of idiomatic Java for Google's cloud platform.

We thought that we could, through open source, more easily sort of evolve the platform to meet special needs for things like ccTLDs, for example, or like restricted TLDs. Open source makes that way easier to do. And we wanted to give back to the community.

So, why releasing this now? Why not a year ago or two years ago or three years ago? Or why not just start out as open source? And initially when we started, we knew that there were things we wanted to, for expediency reasons, depend on Google internal services, which we couldn't open source. We have since worked to get most of those out of the source codes that we could open source, but what we think is some sort of core registry functionality.

We also did a couple of iterations on the model we used for a data store. The first iteration was a little bit like what, I think, a lot of partition data systems do. And we, you know, spent about a year on that model, and we thought we could actually scale a little better by taking sort of a radically different approach to the two, to our data model.

So, if we had open source before we did that sort of scheme and migration, it would have been really painful for people. I mean, this open source things are going to change anyway, but that would have been a pretty significant change, so we wanted to make sure we were comfortable with our model before we decided to kind of open the doors on it.

We wanted to make sure the source code looked pretty, so it's Java coded, so [inaudible] dependency [inaudible]. We use [inaudible] two for that. There are parts of the system that we fulfill, using non-open source components, that you would also, if you wanted to run it. So it's not like a registry in a box, kind of turn key system.

We don't have native protocol end points for EPP or WHOIS, so EPP and WHOIS implementation use HTTP. We run proxies that list port 700 for EPP and port 43 for WHOIS, and do like a HTTP translation for us.

They're pretty simple, but they're running on a proprietary cloud. Donuts actually have their own proxies. They have a version of Nomulus that's running, and there is information about that on our website. And it's pretty easy to write a proxy. So if you wanted to run this system today, you would need to actually provide your own proxy, or you would have to just wait until a proxy was available as part of the open source project, or you could contribute to the project and contribute a proxy.

We have DNS integration, but we do not provide DNS with Nomulus, so we actually separated architecturally the registry from our DNS. We can actually plug in different DNS providers. Our existing provider is a Google internal DNS system. And we have support for other DNS, in a later slide, I'll talk a little more about that.

We have a key store implementation that is not meant to be used for production. So you would need to have a sort of production quality security key locker if you wanted to go and use the system. And we do not have full spec 3 compliant reporting, but we do have reporting that we use to fulfill that goal. And if you don't care about that, then this is not a worry for you.

We chose the Google's cloud because we are lazy and don't want to do administrative work. So it works really well for us, because app engine is super easy to configure. It uses XML files which are very well documented, and for Nomulus, those XML files are exactly the same as they are for any other app engine project. So if you are familiar at all with app engine, this is exactly the same. You don't need to learn any new domain specific language to configure it.

Nomulus itself is written in Java, and you configure Nomulus in Java, with really well documented Java code. So again, no new languages to learn. So, if you're going to use the software anyway, you already have to know Java. So there is nothing new to learn here.

We never have to configure machines, or networks, or firewalls, or anything, because it's all done for us, which is amazing. And we've got built in automatic scaling, which works really well load balancing, which we never have to configure or think about, and security built into the platform.

Google's cloud has a lot of nice administrative tools and dashboards. App engine has its own specific set of tools as well. A really great log viewers. So it has been a huge benefit for us to not have to sort of worry about providing all of those infrastructure as part of the platform. We just get it for free with Cloud.

And we can kick the tires for free on the free daily [inaudible] that GA gives you. So, we have… I mentioned earlier that our initial code base used a different model that didn't scale as well as we thought we could. It scaled similarly to other partition data systems, where you've got a partition based on some dimension.

For a registry that would normally be a TLD, so you have a SRS, and each TLD is their own sort of partition. And we scale that way, so let's say if you wanted to register… Let's say you had a SRS with 100 top level domains, example one through example 100. And you wanted to register test dot example one, test dot example two, all the way through 100.

So, register 100 of these, you could do that concurrently. So all of these could happen at the same time in our original system. And that's, I think, fairly typical of how SRSs are built in those types of partition data systems are built. And we thought we could actually improve upon that, we wanted to scale, not at the TLD level, but at the object level.

And what that means is you can now have, let's say, test one through test 100 dot example one. And test one through test 100 dot example two, all the way through example 100, all concurrently. And that's a significant improvement over our scaling abilities. And that's really what this slide is talking about.

There are some costs of that. I think this slide, unfortunately, is not formatted as well as you would have liked. So, we lost the bottom part of the slide. But there are some costs to this. One of the costs is read only operations. Maybe sometimes, momentarily stale. For any transactional requirements, so they are strongly consistent, so that's not a concern.

And the other two points are slightly amiss, from memory, I believe the other things that we believe are sort of significant costs, but just notable, not necessarily impactfully costs, are some of our operations are necessarily asynchronous. I think those are going to be like contact and host deletes and host renames.

I don't think we've actually ever seen any of these operations in our backend, so I don't know how common they are. But those could be asynchronous. In our first implementation, any of our mutations may have been asynchronous, because we didn't necessarily know how we were going to scale to meet like high demands until we changed to where our distributed APP model.

And the final cost is that getting this right [inaudible] is complicated, so some of these areas of our code base are probably complicated.

So we have a pluggable DNS provider capability, where, out of the box, we support buying via dynamic updates, RFC 2136. And we support Google's cloud DNS through its API. And there are great examples of how to add your own. Any DNS backend provider who has an API can have a plug-in for this, and it works out of the box, I mean, it's really simple.

We have WHOIS and RDAP. Again, RDAP is still new. We are tracking it. We have a WHOIS implementation that, as I said, that's HTTP based, but we have a proxy that runs on port 43. We have a sort of detailed compliant output format, if anyone cares about that.

We do not have full search ability support for WHOIS, but we do have it for RDAP. So, I think adapting that to WHOIS, if someone needed to do it, would be pretty simple. And we have a pretty significant reporting capability. So our data store is data store, actually, cleverly named. And you can do a lot of reporting directly off of that, but it's an object database.

So, in order to make that reporting easier for non-technical people especially, we export our data, both all of the transactions we have, and all of our log in, as flat b query tables. So you can actually use pretty simple SQL to do pretty good reporting.

We can get things like activity reporting, even billing info reports. And we use this actually as inputs toward our billing, as inputs to our billing system, and this is also what we use to do our spec three reporting.

And now we have a video, which is a screen cast. I'll tell you a little bit about what the screen cast is going to show. We'll setup a project, and the project… It looks like we started the video. Can we may be reset that?

Yeah. Sorry. I just want to introduce the video, because I don't think audio does that. Is the audio going to come through? Okay. So, the video is going to show setting up a project. The project name is going to be called Nomulus Demo Alpha. And we call it Alpha because in our build, we have five different environments that get built automatically. We partition our environments by how we deploy them.

We have a production environment, which is just the name of your project with no suffix. Then we have a sandbox environment, which is your project dash sandbox, and that's what we use for [inaudible] environment. We have that same project, dash alpha, which we use for [inaudible] developments.

We have that same environment, dash crash, which we use for like, you know, breaking the system and doing really destructive testing. And we have that same environment named dash local, which is a local version of the application which you can run on your desktop. It requires quite a bit of resources, but it can [inaudible] written around trip time if you're doing some testing.

Okay, now we can have a video.

VIDEO: Welcome to the Nomulus demo. We're going to go ahead and create a Google cloud project to get started, and we'll call ours Nomulus Demo Alpha. And once that's created, we're going to go ahead and setup some storage buckets, and these are described in our architecture documents.

And I'll take you through the process of creating one of these, there are quite a few so we won't create them all here. I won't bore you to death. We're going to prefix the name of the bucket with the name of the project, which is for us, Nomulus Demo Alpha. And we'll create our billing bucket.

And then we'll just go ahead and, through the magic of video, speed to the end so you'll see all of the different buckets that get created. And here they are. Now, we're going to go ahead and start a cloud shell where we'll use our command line. And the first thing we're going to do is install [Basel], which is our bill system.

In order to do that, we're going to refine the local repository database to include Google special [Basel] repository, so we can just use an ATP get command to install [Basel], which is really nice. And once that's done, we'll install the Cloud SDK.

And we'll need that to do our deployments later on.

And there we have it. So now we can go ahead and retrieve the source code. We'll create a directory called Nomulus for that. And we're going to use a get command to pull it down from the get at repo. You do not need a get account to do this, you can just go ahead and use gift from the command line.

[Inaudible] and just do a [inaudible] command, which is what we're going to do here. And it will pull down the source code.

Here we go. And now we're going to go ahead and configure the system. So we're going to configure some Java files, and some XML files, and we'll start with the Java file. We're just going to change the application names, really just for aesthetics, but we're going to go ahead and do it anyway.

And we'll just call ours Nomulus Demo.

And once that's done, we're going to go ahead and configure our sample package, which is also a Java file. And here, the most important thing that we're going to do is just change the project name to be a prefix of our project name which for us is, Nomulus demo alpha, so we're calling ours Nomus demo.

And if you'll notice down here, the environment, which is our alpha environment, will be automatically added in the code.

And once that's finished, we have to go and configure some XML files, which are just normal [inaudible] project files. And we'll have to do once for each of the backends, and we have three backends. We're going to do this for the backend module, the default module, and the tools module. And again, we're just changing the application tag here, and we're changing that just for aesthetics, but we're going to go ahead and do it anyway.

So there is one.

Here is the second.

And here is the third.

Okay?

Now that that's done, we're going to go ahead and build Nomulus.

And this is just going to be a [Basel] build command, that's going to build all of our modules, but if you wanted to, you could build just one module. You could even deploy just one module, but we're going to go ahead and do all three here.

And that wasn't that fast in real life, but again, the magic of video, and these are the files that it creates.

We're going to unzip he environment we want, which for us is the alpha environment into a deployment directory.

And once that's done, we're going to implement a temporary workaround, by the time you see this, may no longer be necessary, but we have some restrictive ACLs on our tools, and we're going to just remove those restrictions, so I can run our admin tool in our demo environment.

And that's just going to be done by commenting out the restrictions here. And as I said, this may no longer be necessary by the time you see this video.

And there is another temporary workaround. We're going to go ahead and add a modification to our Cloud environment to use local authorization.

And here we are deploying our project.

And this is going to deploy, as I said, all three of our modules, which could only deploy one at a time if you wish. And this takes a little bit of time. We are speeding this up, so this is roughly two times as fast as it would be in real life for me, and this is on a Cloud shell which is a little small [inaudible] that gets created for you automatically.

So, how fast or slow this is, is going to depend on your local environment. And once that's done, we're going to go ahead and use one of the artifacts from the build, which is our Nomulus command line client, to interact with the system. And it looks like our final modules is being deployed now, which is the tools module.

And it looks like it's completed. And there we are.

So, now we're going to go ahead and create a registrar using the Nomulus command line tool.

And that just interacts directly with our backend. You can see the modifications it's proposing here. Then we're going to create a registrar contact to use for the registrar so we can access the counsel. And I'm going to create a top level domain called example.

Now we're going to create a host, and a registrar, I'm sorry, a regular contact we can use to create a domain with. And that's using EPP commands, as you can see.

Here we have a 1,000 return code, which is good. Now we have a host, and we have a contact, we're going to update our registrar to get access to the example top level domain, and we're going to create a domain.

And then we can do a WHOIS lookup using our command line tool, and that is the WHOIS output. Now we can go ahead and look at our counsel. So we're going to [inaudible] counsel. This is what the page looks like, and the URL to get there is just the project name [inaudible] dot com, and then slash registrar.

We have different upside tabs, we have WHOIS contacts, security. We have billing information, and a contact us page. Now let's take a quick look at the Cloud dashboard. Let's actually do a WHOIS lookup. We can do a WHOIS lookup from the web as well, and it's the same output we saw from the command line.

Now we're going to switch back to our Cloud dashboard, and we'll take a look at what those tools look like. This is the main Cloud dashboard which is an overview of all of the different elements of the Cloud that you have, and things like traffic and so forth.

Building information. And we're going to take a look now at the app engine portion of the dashboard, which is a little more interesting for us. There are different things you can see here. Just different services, versions, and instances, task views, if you notice the task view share are pretty much empty.

There are some other things in the sidebar. That's a little overview. And we can now go and run our load test, which is built into the system, it did that. We're going to use something called Postman. You can just use the URL if you want to, but this makes it a little bit nicer. You can enumerate the parameters here.

We're going to click to run button, and in real life, again, this is going to take a bit longer than that, depending on your parameters, but we'll speed it up with video. Now that that's completed, we're going to switch back to the [inaudible] dashboard, and we're going to update our view of the traffic, and you can see a little spike there from our load test.

And that's the one hour view of that spike.

And go look at our task views, you'll see now there is a lot of activity here where there are all zero before.

And we can take a look at our logs, and these are things that are really useful for debugging, but here you see a lot of different log activity. You can take a look at data store directly, and there is different ways to do this. We'll just use the little dropdown to take a look at the domains we created, and you can see here there is a lot of data. And here is the little funky domain names we created automatically to load test.

And we automatically export metrics into big query, so let's take a look at that. This is the big query [inaudible] metrics table, and we can query that, and we'll run a query that will give us a distribution of, let's say, the create command for different roundtrip times that you're basically latency milliseconds. And that's the distribution you should see lots of [inaudible] and lots of distributions across different latencies.

And we'll do the same for the input command. And you see a lot more of these input commands are bunched up towards a single digit, millisecond latencies here for info. And we can automatically export this to Google Sheets, so let's go ahead and do that. And once we're in Google Sheets, it's really easy to just craft this, so we'll get a view of our latency by distribution, and you'll see the redline is the distribution. And most of the commands are at the very lowest latency.

But this is sort of a long tail, where we've got like single digit distributions on high [inaudible], but the majority is commands or [inaudible]. And there we go.

RICHARD ROBERTO: Like watching a Billy Idol video.

So now we go back to the slides. And just to clarify, I noticed having just watched this video now, and I've avoided doing that, I mentioned things like buckets, I'm sorry, backends and modules, and I used both terms interchangeably. They used to be called one, now they're called the other, and I forget which is which, and I apologize for [inaudible] those terms.

So they're the same thing in the video. So, for more information on Nomulus, we have our website at Nomulus dot [foo]. It currently just redirects to our Get Hub. We've got Java Doc, that's actually really useful. We've got some other documentation, which is inclusive of our installation guide, and our configuration guide, and our architecture guide, and some overview documents. Even just the read me from the Get Hub site, it's pretty useful.

We have a discussion group, which I encourage people to go ahead and join if you're curious about Nomulus. If you're interested in contributing, that's the best way to get involved. And now we have time for Q&A.

EBERHARD LISSE: I have one question. [Inaudible] query, the database that you're using, I've never heard about that, but I don't [inaudible] more or less. I've looked it up. I like to do my reports in L, because it can, it has modules that you can interface with databases, and it can write your reports in [La-tech].

So, there is a module for that database. Can you talk a little bit about this database?

RICHARD ROBERTO: I'm afraid I know very little about [big?] query. We use it, I use it because it's just a simpler way to interact with the data. I just use like pretty straightforward SQL, so it is different sort of reporting and visualization software that integrates with it, but I don't use any of that stuff. I just use the command line.

EBERHARD LISSE: Diego?

DIEGO: [Inaudible]. I have two questions. One is, do you take into account DNSSEC in this development? And the other question, is it possible to restore this in a hosted environment? Because what I see depends on Google Cloud.

RICHARD ROBERTO: So, I didn't catch the first question.

DIEGO: DNSSEC. DNSSEC.

RICHARD ROBERTO: What is your question about DNSSEC?

DIEGO: Okay. If you take into account implement in the DNSSEC in the two, or is there some way to implement DNSSEC?

RICHARD ROBERTO: The tool does not implement DNS. We don't have DNS. So, we've separated DNS from our system.

DIEGO: Okay.

EBERHARD LISSE: Yeah, but the key management. If the registrant needs to upload a key [CROSSTALK]…

RICHARD ROBERTO: That's part of like standard EPP, so yes, we support that.

DIEGO: Okay. And the other question, is it possible to have this environment hosted in a local?

RICHARD ROBERTO: Well, as part of a test environment, you can run an app instance locally, but I wouldn't run a production service off of that. It's very tightly coupled with Google Cloud services. I mean, it's theoretically possible that there is an open version of like app engine. I think they're probably are some. I don't know how complete they are.

But one of the reasons we chose the Cloud is because we didn't want to have to deal with all of the different things you need to do to run your own system. We really are much lazier than people think. Yes?

UNKNOWN SPEAKER: [Inaudible]. My question was answered already. So, maybe, do you have some plans how to change it? How to allow people to install it locally? Or to replace the Google by [inaudible] or anything?

RICHARD ROBERTO: I have no plans to do that, no, but I mean, it's open source, so there is no reason someone else couldn't do that. We do use Objectify in the source code though, so we have an object interfaced to our data store. I think that because of our reliance of Objectify, it's theoretical that you can have, you know, Objectify be this sort of abstraction layer, and have Objectify talk to any backend, but I don't know how simple that is, because I don't know how complicated the Objectify interface part is.

So, it sounds like it should be possible to do, as long as Objectify works with whatever is underneath it. But I don't currently have any plans to do that.

UNKNOWN SPEAKER: Okay, thank you.

RICHARD ROBERTO: Yeah.

EBERHARD LISSE: Are there any other questions?

Please feel free, we have got lots of time.

LARS LIMAN: Hi, Lars Liman from Net Node. How do you foresee the continuation of this project? Open source, yes, but do you seek input from the community? Will you ask contributions, code contributions? Or is this a Google project which you just leave open so people can work from it?

RICHARD ROBERTO: So, I think there is probably three different ways we kind of envision this working. One is, people just looking at the source code because they're curious. Like I said earlier, seeing how we decided to answer some unanswered questions as part of the spec, you know, we have open specs, but there is a difference between an open spec and an open implementation. We have now, yet another open implementation that has its own take like policies, some ambiguous parts of RFCs.

So, we expect that some people would just do that part, and then decide whether or not they're going to adopt that in their own source code. It has got nothing to do with ours, right? We think there is also people who would just wait for the system to be updated by whoever, and as it gain features, they'll adopt it and fork it off into their own versions.

And we think there is going to be people who want to join the community, and give back. We're actually really eager to get people to do the latter.

LARS LIMAN: Would you, in that case, coordinate this effort and receive input, co-contributions, into the open source project that you provide?

RICHARD ROBERTO: Yes, absolutely. So we have a… And that's what I mentioned, if we go back a slide. So, joining that discussion group is the easiest way to start getting involved, and all you need to do is say, hi, I'm interested in contributing to the project. Where do I begin? And someone in our team will be in touch.

And there is different things that we think are useful, but if you have your own itch to scratch, it's open source. We do plan on being judicious about what we allow in the repository. We don't want it to become a playground, but we have every intention of taking contributions. We want that.

LARS LIMAN: Okay. Thank you.

RICHARD ROBERTO: Yeah.

EBERHARD LISSE: No more questions? Okay, thank you very much. You can give him a hand. [APPLAUSE]

And then I turn to the left, and [inaudible] from [inaudible] and his colleague, whose name I didn't catch, who will introduce himself just now, will speak about the virtual keyboard.

UNKNOWN SPEAKER: Yes. Hi. The topic of this session is, IDN and the registry services. And, our presentation will combine both. I will talk about the dot [inaudible] registry, and its peculiar IDN table. And virtual keyboard that serves as… That will help the registrars and registrants.

My name is [inaudible], and next to me is [inaudible], who played a major role in the development of the IDN table for the dot [inaudible] registry, and also in the virtual keyboard it serves.

So, at first, I will present some motivation, why we thought the development of the virtual keyboard would be a good idea, and will be helpful for the community. Then I will shortly present the virtual keyboard itself, give a short overview.

And in the end, I will show some practical examples, how the virtual keyboard works, and how it can be used. So, first motivation, the [inaudible] script is distributed all over the world. There are actually more than 600 million people living in countries where at least one language is spoken that users [Arabic?] script.

And this lets [inaudible] to the idea to create a new top level domain, called dot [inaudible] in [Arabic] script, of course. That is not based on one single language, but which is actually based on the [Arabic] script. And therefore, it can be used by all of those people even though they speak different languages.

And so if you want to register such a dot [inaudible] domain, you could use the control panel of the registry, [look?] in there, and then go to the domain create dialogue, and somehow you need to type the domain name you want to register.

You could of course, use the A label or Punycode notation, that means it's [N, N?] dash, dash, but of course, no one really knows how the domain name looks like that in that notation. So, you would actually use a Unicode version, then the domain names will look like this, but the problem will be how to type this if you only got an English or similar keyboard available, you just don't have the keys necessary to type those [Arabic] letters.

And even if you had hardware [inaudible] keyboard, chances are that are not all of the keys that are available in the dot [inaudible] registry will be available on that keyboard, since to my knowledge, there is currently no universal [Arabic] keyboard available, and not even close to that.

So, that was one reason we thought that a virtual keyboard would be helpful to actually type those domain names. The second motivation, the second reason, is to provide the user with domain label [inaudible] information. If you run a registry which allows second level IDNs, then you'll need to specify a so-called IDN table, which is then hosted in the [inaudible] repository, and just accessible to everyone.

I put in here the link to the dot [inaudible] IDN table. It's still an old version, 1.0, but now it has been updated to a newer version, but I didn't have time to update the presentation itself. So, an IDN table explains what letters are resolved, and in what context they are resolved.

For a simple IDN table like for the German language, you just add some more letters like the [inaudible] we have, and that's it. The letters can be used wherever you want them to be, and there is no restriction. This is different for the [Arabic], because the letters there have a so-called joining property. This means that depending on what's before and after the letter, it can look different.

Two letters can join, and then they can have a totally different shape. And that's the reason why the [Arabic] IDN table is quite complex. You will need to define which are allowed variants, and which is a [inaudible] form, because some letters might look the same way, depending on their context, and therefore, it's important to make sure that not two different people can register two different domain names, but which actually looks the same.

That's why we introduced the variants, and the [canonical?] forms. So, to give you an example of such a rule in the Arabic IDN table, I just wrote down here the rule for the so-called [inaudible] group, which is a character of the Arabic script. And this [inaudible] group, it's separated in two paths.

First, if you have a character which has an Unicode point 6CC, for example, then we do not care what character comes before that character, but depending on the character which follows it, the variants and the [canonical] form, or index form, will be different. If the character is followed by a character, which has a right or duel joining property, which means that this following character can join with the character before, to the right, or to both sides, since we write from right to left, this means that this character is able to join with a 6CC character.

And in that context, the 6CC and the 64A characters, can be used interchangeably. There are variants of each other, but if on the other hand, the 6CC character is followed by a character that is not possible to join with it, then the 6CC and the 64A characters are variants of each other and can be used interchangeably. So you'll see the rules are not so simple, if you have not just, this one rule, but several rules, it's very difficult for people with a background that is not so technical, and is not so much in computer science, then [inaudible].

It will be difficult to see what variants are possible for my domain name. And that's the second reason we thought the virtual keyboard would be of help. So, let's take a look at the virtual keyboard. This is total look, if you access it via the webpage, keyboard dot nic dot [inaudible] in [Arabic] script, since you would need to be able to type [inaudible] in the [Arabic] to access this with [inaudible] and domain that is ASCII only, that would be s dot [inaudible] dot [inaudible].

And you will see several letters at the top. The digits, as we use them in English. If you hit the shift key, you'll get different letters and all sort of different set of digits. And with the lock key, you'll get them yet again, different letters and a third set of digits.

So, let's see how it works with some practical examples. For this, we have selected three domain labels, because they all have some interesting properties regarding the variants. The first one is [inaudible], which is internationally name of a German state. The second one is [inaudible], which is [Farsi] and means [hazel?].

And the last one is [inaudible], which is [inaudible] and means [food?], at least I've been told it means that. I hope no one played a trick on me there. So, now I would like to switch to my screen, so I can show you my keyboard version to actually type something.

There it goes. Let's start with [inaudible] label. So I type a B, an A, a V, an A, R, [E?] and A. Though it's a bit slow here, sorry for that. Okay. So, what you see now, on the left below the keyboard, is the [A?] label version of that domain. It also tells you how much, how many characters the [A] label version has this big important, because even though the actual [Arabic] word only has seven characters, the [A?] label version already has 16 characters, then as you probably know, each label may only have up to 63 characters.

So, this is an easy way to check whether domain name you want to register is short enough to fit this limit. On the right side, you see the [U] label version, and below that, you see the reference view of the U label version.

You might ask, why do I need a reference view? This looks exactly the same as the U label version, but if it looks the same, then everything is good. And you don't need the reference. But I have actually used some computers, where it does not look the same, because some fonts seem to have some problems with [Arabic] letters, and sometimes, they are missing a letter or two, or sometimes they do not apply the joining properties correctly.

So, this reference view is rendered on our server, then supplied as an image. So this will always be the correct version. Now, below that, you see all of the characters, from right to left, and if there is a second character below this, this is possible variant for this character.

Now it's a bit small to see, but we have here this character six for A, which I talked about, the [inaudible] character. And it's the variant is 6CC, the [Farsi] [inaudible], but this is only because the following character has a possibility to join with this character.

If I now remove this character, then you will see that the variant also disappears. So, the second example was [inaudible], which is written like this. Well, still looking behind.

Okay. Now the M, the [inaudible], the [inaudible] character and second [inaudible], the interesting thing here to notice is that the second and the fourth character are actually the same, namely the [Farsi] [inaudible] character, but their two variants are different, because for the second character, it's possible to join with a third character, and for the fourth, there is no character to join.

And the last example, that is the [inaudible]. The word itself, only has three characters, but you can see that the first character has one possible variant, and the second character even has two possible variants. So in total, this word, there are six variants for this word. That's the reason we did not right each variant, but only the possible variant for each character because this could get rather tired, due to the multiplication.

One other nice thing is, the URL of the keyboard always reflects the word which has just been typed, so you can send a link around to someone else to show this label. Yeah. That was my practical presentation.

So if there are any questions?

EBERHARD LISSE: Warren first.

WARREN: Warren [inaudible], Google. Apologies if you covered this, I got sidetracked reading email. Your example showed this working in a web browser, the bit that I missed was, is this mainly being done by Java Script? Or are things actually sent back to your server and then done there and sent back, stuff in the box?

UNKNOWN SPEAKER: The logic of the rules, are actually calculated on the server. So it goes back to the server. Because the same rules are also implemented in the registry software itself, it's the same Java package, and we share the code for this also. So both are consistent.

EBERHARD LISSE: Okay. I must say, this… I never actually considered that such a thing was possible, or even needed. But the most I encounter are the German [inaudible]. I think this is a rather clever idea, and I'm quite impressed with it. And my only question is, is this proprietary? Or will you make the source code available?

UNKNOWN SPEAKER: The source code is not available, but the keyboard is free to use in your website. So if registrars would like to have this feature for their registrants, they can simply use this keyboard within their webpages.

EBERHARD LISSE: And another question, will this only work on dot [inaudible], or will this work on every Arabic script top level?

UNKNOWN SPEAKER: If you just want to type the letters, it will, of course, work for all Arabic words, but the rules, which determine which are variants and which are not variants, they are specific for the dot [inaudible] registry. [Inaudible] was one of the major people who created these rules. The thing was, since we cover many languages like Arabic, Udo, Persian, Kurdish, and Malay, this set of rules tend to be created to cover all those languages.

The other Arabic TLDs are just concentrating on one language, and therefore they have fewer characters available, and not so complex rules are needed.

EBERHARD LISSE: Okay. Thank you very much. I think we should give him a hand. [APPLAUSE]

Next is Christian [inaudible]. I think the presenters, you can, if you want, vacant the premises. And go and do your email. Warren, you are a member of this group. You should not say these things.

[SPEAKER OFF MICROPHONE]

Not say them. Okay. Next one will be Christian [inaudible], and he will speak about the valley box. We are not strapped for time, so we're not in a hurry. Take your time.

CHRISTIAN: Okay. Thank you Eberhard. So my presentation will not be that long, I think. Can you make it bigger?

Okay, great, thank you. So, this presentation is about the valley box, which is this tiny device that I'll be talking about. And it's about bringing the NSEC validation into the home. This is work that was done by [inaudible] who is with SIDN Labs, and I'm with [inaudible], and I'm speaking on behalf of [Yelta?] here at the tech day. SDIN is the registry for dot NL, that's the country code of the Netherlands. And that's a small country actually in Northern Europe, and we're boarding with Germany, just in case, if you want to look it up on the map.

SIDN labs is our research team is working on several projects including this one. So, DNSSEC validation, you guys all know that DNSSEC is basically about two parts. One is about DNSSEC signing, and the other one is DNSSEC validation. Signing is about adding signatures to domain names, digital signatures.

And validation is about validating those signatures in order to assess if a certain DNS query was authentic or not. And also, as you guys probably also know, DNSSEC signing has been a considerable success. So, for example, for dot NL, we have 5.6 million domain names in our registry, and 2.5 million of those, roughly have been signed with DNSSEC. So that's roughly 45% of the zone.

So, that's a pretty good achievement. And there are similar zones that have similar, there are similar ccTLDs that have achieved a similar DNSSEC signing rate, such as the new regions, and also in the Czech Republic. The thing that's missing though, at least in the Netherlands, is validation of DNSSEC signatures.

And as you can see, or maybe you cannot see it, but the Netherlands is extremely deep red in this graph, and the, let's say, the leading countries are actually Norway and Sweden, where DNSSEC validation is green. So that's very good.

So, but DNSSEC validation has not seen much uptake in the Netherlands, and that's primarily because it depends on the ISPs to do something. They need to turn on DNSSEC validation on their resolvers in their network. And they're kind of reluctant to do so for two reasons.

One is they think it's difficult and hard to do, and the other one, other reason is that they think that it will lead to support calls for their support desk, which directly translates into costs. So for example, the thing that they worry about a lot is, if there is an error in the signature, for example, over the this DNSSEC signed domain name, then, and they validate on their network, then that domain name will no longer work for their customers.

Whereas, it will still work for customers of their competitors who do not do DNSSEC validation. So, for us, that's bad news, because we have 45% of our zone signed, so that's a huge security potential, but it's not really being made use of very much at this point. So since we can't really rely on the ISPs to do something in the short-term, we decided to investigate if we could bring DNSSEC validation to home networks.

So the ultimate goal, of course, would be to have DNSSEC validation running on your end device, like laptops, and iPads, and that sort of thing, but that's kind of difficult because most users don't know much about DNS let along about DNSSEC.

So, we were looking for something that would bring DNSSEC validation to the home in a very user friendly way. And that's why we started to look at devices like these, which is a small router that runs on open W RT, which is an operating system very often use for these tiny devices. And what we did, was we took the resolver of [inaudible] labs, which is unbound, extended it with a few features, which I'll be talking about in a minute…

The mics are still not working right. All right. We're back online. So, let me catch up here.

So, we were looking for a user friendly way to bring DNSSEC validation to the home, looking at devices like these that work on the open WRT operating system. So, what we did is we took the unbound resolver of [inaudible] labs, and extended it with a few features.

And we basically put it in an image that you can install on open WRT devices. So, if you do it with this device, for example, you simply plug it into your network, it will create a new SSID, so a new wireless LAN, and then you basically connect through the internet through that wireless LAN and you will get DNSSEC validation out of the box, through this device.

So that's something that we did. And note that the ultimate goal was not really about the hardware, but it's about the software component that we put on the device. So, our ultimate objective is to make the software available for manufacturers of devices like these. And which is why we open sourced it. I'll show the URL later on.

We open source this software to make it available to CP manufacturers to not only… So, we not only made DNSSEC validation easily available for end users, but also for CP manufacturers to put on their devices.

And what we did with the unbalanced software that we took from [inaudible] labs, so [n bound?] is an open source validating resolver, and we extended it with two features. One is we improved error reporting, so we… So, you can see on the right hand of the slide, that if there is an error in the signature, for example, of a certain DNSSEC signed domain name, then we will provide more information on the error that occurred, let's say that in addition to the, let's say, default error that you would normally get.

So that users know what's going on. And also, we added a user friendly NTA management. NTA stands for Negative Trust Anchor, which means that if a certain domain name does not validate, you can basically put an exception on it and still go to that webpage. So that's like clicking away a security warning in a browser, which of course, reduces security, but we think that this is a justifiable approach because the number of signed domain names, sorry.

The number of domain names with a validation error in their, is pretty low. At least in the dot NL zone, we have 2.5 million signed domain names, and roughly 1700 of them returned validation error. So that's not too many. And also, this takes away the concerns of the ISPs that, let's say, they're not… Excuse me?

[SPEAKER OFF MICROPHONE]

About 10 minutes, okay. I'm sorry. I was unaware that the slides were not up there. My apologies.

If it takes 10 minutes, I propose we go on, because…

[SPEAKER OFF MICROPHONE]

Yeah. Yeah, sorry about that.

EBERHARD LISSE: Everybody can just go on the echo, the stream that we have, and see it on that screen, because it is, should be displayed there. But I have been informed by my son, who is watching it remotely, that he can't see the back screen either.

CHRISTIAN: Okay, so the addition of user friendly negative trust anchor management, to set exception on domain names that do not validate. So, these are the two features that we added. Like I said, we open sourced the software so that it would be available for everyone to use. So, these are the URLs. The developer's guide is available from our website, Valley Box dot SIDN Labs dot NL.

And the actual software is available from our Get Hub repository. And in summary, we believe that we have developed a means for non-technical people to get DNSSEC validation into their home network, as well as the means for CPE manufacturers to get DNSSEC validation on their open W RT devices more easily, through the open source component.

We've added features such as better DNSSEC error reporting and negative trust anchor management. And we also believe that this kind of, these kinds of devices are a platform for experimenting with additional features that we are working on within SIDN Labs. So, for example, we have a system in which we monitor the newly registered domain names under dot NL and check if the DNS traffic that they receive is abnormal, or if it shows certain, you know, phishing like patterns.

And if we can detect that kind of information, we could potentially push that information on those domain names to devices like these, to warn end users so that they're visiting a potential harmful site.

And finally, if you're a technical person, you can of course, buy your own [inaudible] device, flash it with the software image that we provide on our website, and then give it to your friends as a gift.

So, that was my last slide. If there are any questions, then I'll be glad to take them now.

EBERHARD LISSE: The post is two Christians in a particular order, I have thought about a lot. Can I have one? And how much does it cost?

CHRISTIAN: Yes, and they cost about 40 to 50 Euros.

EBERHARD LISSE: That's not the… So yes was the answer to my first question. Can I have that one?

CHRISTIAN: Yes you can, but there is no plug or anything like that.

EBERHARD LISSE: Don't worry [CROSSTALK], we'll figure out plugs don't worry. But anyway, the real question is, have you go a few more? I see [inaudible] coming to the microphone, so I don't have to mention that he has got four ATLAS probes to hand out, and he asked me to say that. Where do we get them if I want to pay for it, and how does this work?

CHRISTIAN: Actually, it's a commercially available device. You can just buy it at your favorite hardware store.

EBERHARD LISSE: So I just buy a device, and download your software, and figure out how to get it in? Okay.

CHRISTIAN: There you go, yeah.

EBERHARD LISSE: Last question from me, uninformed technologist, is, on your website, exactly what device do you need to buy where you can order it and so one?

CHRISTIAN: Yes, yes. Actually, it depends on the operating system, not so much on the device. So, if other open WRT devices, you'd also be able to run it. [CROSSTALK]

UNKNOWN SPEAKER: You already answered one question. This is [inaudible] from NL dot Labs. A follow-up question and that is, are you willing to give the software changes you made Unbound back to NL Labs, so it can stack it in the [inaudible] directory?

CHRISTIAN: Yeah, I would see, I don't see why not.

UNKNOWN SPEAKER: So people might actually use it and change it for other devices?

CHRISTIAN: Yeah. We haven't talked to you guys about that, but yes.

UNKNOWN SPEAKER: Okay.

CHRISTIAN: Okay.

EBERHARD LISSE: I just wanted to point out again, that's the guy who has got four Atlas devices to give away, if you want one.

UNKNOWN SPEAKER: Find me.

UNKNOWN SPEAKER: This is [inaudible] from [inaudible]. One question. You said this is based on [amount?], so I believe this is a real RFC 1511, but is this true?

CHRISTIAN: Could you repeat that? I didn't get it.

UNKNOWN SPEAKER: RFC 1511, that is automatic trust anchor update, because root key rollover will come next year. So, it should, it have to comply to KSK rollover, [inaudible] KSK rollover. So my question is…

[SPEAKER OFF MICROPHONE]

CHRISTIAN: I'm sorry, I don't know. You need to ask [inaudible], he knows that sort of thing, yeah. You can drop an email. His contact details are on the…

EBERHARD LISSE: On the agenda, that is downloadable, the presenter's name are highlighted and clickable. So, if you want to send an email, just go to the agenda, click on the name of the presenter, and you're email… You know how this works, obviously.

But then, if somebody has questions to presenter, that's an easy way of contacting them.

ANDREW: Hi. Andrew [inaudible], ICANN. So if validation fails, the user gets some kind of prompt or something, like the user gets redirected to some kind of webpage where they get to input this negative trust anchor?

CHRISTIAN: Yeah.

ANDREW: What do they see? Was that in the slides?

CHRISTIAN: That was in the slide. That's what you said.

EBERHARD LISSE: On the slide, when the slide wasn't working.

CHRISTIAN: Oh, okay. Sorry about that.

ANDREW: What happens if it's not a HTTP request that caused the validation to fail? So, I mean like, what if there is not like a user behind that DNS query? So like it's some IOT, like it's a light bulb doing a DNS request.

CHRISTIAN: I don't know. So obviously, a light bulb cannot press a button or something. So, this was meant for end users to use. Yeah.

ANDREW: So if I go to like CNN dot com, and it loads Java Script from 20 different sites and one of those fails validation, would I get that?

CHRISTIAN: Probably. I must admit, I don't know all of the details, so if you have any follow-up questions, you can contact [inaudible] on that.

ANDREW: Okay, cool. Thanks.

CHRISTIAN: Thanks.

UNKNOWN SPEAKER: Well, this is, since this is a CPE [inaudible], so if you have, if your light bulb stops working, I mean, you will see that [inaudible]… You'll see, you want see anything, but a part from that, I mean, it's, you can always install in this CP [inaudible] and can… Light bulb itself doesn't need to do it. That's because [inaudible] you have connected that in your own [inaudible] network, but it's also fairly unlikely that anybody, where the light bulb talks to, will have DNSSEC in the first place [inaudible] security measures.

EBERHARD LISSE: [Inaudible]

UNKNOWN SPEAKER: Really cool stuff. You know, this is the kind of stuff I love seeing coming out of labs. It's wonderful. How big is the footprint for just the validator part? I assume you're providing a full firmware update to open WRT, but what part of that, DNSSEC part in this part of it, how many K is that approximately?

CHRISTIAN: I don't know. You would have to ask [inaudible]. So I'm proxy for him now. So all the technical details, please drop an email, and he'll be glad to respond.

UNKNOWN SPEAKER: I didn't mean to put you on the spot. This is really great stuff. But sometimes size matters, thank you.

EBERHARD LISSE: All right. I think we should give him a hand. [APPLAUSE]

And that means we're going to have lunch. We must be here at about 10 past one, so that we can start on time with the host presentation. Thank you very much.

[END OF TRANSCRIPTION]