MONTREAL – Emerging Identifier Technology
Tuesday, November 5, 2019 – 13:30 to 15:00 EDT
ICANN66 | Montréal, Canada

ADIEL AKPLOGAN: Welcome everyone to this fourth Emerging Identifier Technology session at ICANN meeting. To set a little bit the background of this session, we had started a few years ago to host this session at ICANN meeting with two key objectives.

The first one is to provide the community in general insights on some of those emerging technologies in order to help people attending ICANN meeting and are concerned about identifier general and when it comes to internet to know what is going on out there. But it's also help us from the office of the CTO to identify some of those identifiers and deep dive into them, study them, and provide more detail information to the executive team to the Board on probable impact on the identifier system in general and the DNS, particularly.

For those who have looked at our strategy document, there is a specific point about that in the new Strategy Goal 3-2B that actually requests, ICANN to have a mechanism to look into new technology and when appropriate embrace them. So this gives us kind of a way of looking into those by also inviting those who are actually working on those technologies to come here and tell us what they are doing and have the opportunity to exchange with the community. So this is the fourth one.

We had, if you want to read the previous slide, you can also see them, ICANN58, 60, and 64, where different technology has been talked about like name coin, DOA, and such. So, for this session, particularly, we're going to talk about two emerging identifiers. Some of us have read about it. One is the GNU Name System and the second one is Handshake. The GNU is presented as a decentralized domain name system instead of using URT, it's using direct graph, and we'll hear more about that.

And then we'll have a presentation from Paul Hoffman from the office of the CTO that will kind of give us a little bit more background on the identifier system and what has happened in the past, what have you seen as evolution in that area in the past, and how can we see those new emerging identifiers play out in what ICANN does in general.

So the way we are going to run this, we will have the presentation first. Each presenter will have around 20 minutes to present their topic. Then we'll have questions at the end. So please hold your questions, write them down, and we'll have 40 minutes approximately at the end to interact with the speaker and ask questions. Will that work for you? Perfect.

So the first presenter today is Martin Schanzenbach, he is IT security researcher at Fraunhofer Institute for Applied and Integrated Security in Munich. His area of interest is secure and decentralized system and digital identifiers. So he will tell us a little bit more about his work with GNU nets on the GNU name system. Martin, the floor is yours.

MARTIN SCHANZENBACH: Thank you very much for inviting me to Montreal to present on the new name system. Maybe, first of all, GNU itself is essentially a protocol stack to develop decentralized applications, So it's from network protocols to transport protocols to applications. It's supposed to be a stack that enables developers to develop applications in this regard and those applications also need a name system.

So very early on we thought about how this naming could work and obviously DNS was also initially an option. But DNS, in a nutshell, in order to explain DNS, the first thing that I would like to explain is why we actually created our own name system. And the reason for that, is that from a security researcher's perspective, DNS is broken in various ways.

So first of all, it still remains a source for traffic amplification. It enables DNS censorship which occasionally causes collateral damage even in other countries. It is part of the mass surveillance apparatus and it is abused for offensive cyber war. Now the existing patchworks like DoT/DoH, even DNSSEC and DPRIVE, do not fix this. And that is why we said we need a blank slate and we need to implement a name service from the ground up. The new name system is a fully decentralized name system.

Names in GNS are not are not inherently global, but it supports globally unique and secure identification. It features query and

response privacy and provides a public key infrastructure. So, in GNS each zone is associated with a cryptographic key pair and delegation between zones is supposed to establish a trust relationship. This point is very similar to how a DNSSEC works.

What's interesting about GNS is that it's interoperable with DNS. So you can use GNS side by side with DNS. You can use it instead of DNS, or you can use it standalone. Last year we executed some usability studies in order to see if users could actually tell the difference whether or not they were using GNS or DNS, which you can find on our project page. And what we found is that to the user it doesn't really make any difference. And that was also the goal.

GNS is already now six to seven years old, and this is why there already exist a number of applications that use it, first and foremost, for example, re:claimID, which is used as a self sovereign digital identity management system, which could be used as an EID system, as well. SecuShare, which tries to implement the social network and use GNS to discover social places.

And also within the scope of the project we did last year, we implemented a healthcare use case where patient data was exchanged between insurance companies and private persons. And finally, of course, GNS can also be used for what DNS is used in most cases, and that is host addressing. Because GNS tries to be interoperable with DNS, it also keeps the resource record format of DNS, so all the resource records in DNS can also be found in GNS, and more.

So in order to understand what is different inherently from DNS and GNS, I will go into very deep technical detail, but I will show you the fundamental differences between them. So first of all, in GNS all zones and resource records are stored in a distributed hash table.

Now with the emergence of blockchain you might have heard of distributed hash tables, because they are often used in the form of IPFS to efficiently store large amounts of data. A distributed hash table allows us generically to map any key to a value.

Now usually you would simply map a domain name to a resource record set or to a simple resource record, such as example.com maps to an IP address. Now obviously this would not be ideal because first of all, this value could be overwritten and second of all, any observer that observes movements in the distributed hash table could observe who is querying what name, and also what names exist in a zone, which is something that DNS from the beginning, tried to avoid. So GNS in order to store records implements a private information retrieval scheme.

Now at this point, usually I would show you some slides using the cryptographic implementation of this private information retrieval scheme but you will just have to believe me that it implements one. In general, a private information retrieval scheme is a protocol that allows a user to retrieve an item from a server in possession of a database without revealing which item is retrieved.

Now in GNS, the server in possession of a database is simply the DHT. This allows us to implement query privacy. So queries and also responses do not reveal what domain is resolved or what resource records are contained in the response. The records themselves are encrypted and signed by the zone owner. They can be decrypted by the resolver that has been used to query for the record by deriving a symmetric key from the queried name and the public zone key.

Now the query privacy and record confidentiality property together provide us with zone privacy. So a large problem for a long time in DNSSEC was that the enumeration of the zone was possible through kind of a trick that could be used for nonexistent records, as you might know. But in GNS the only way that you can actually verify that a name exists in a zone is by querying it, so inherently you must be already aware of the name.

So zones cannot be enumerated because that's what you would do in order to learn with names exist in the zone. It also provides censorship and DDoS resistance through two distinct features, one feature is that the DHT provides us with inherent decentralization, so it is a resilient directory.

It's very difficult to DDoS exactly those sets of nodes that make up a specific zone. The other feature that prevents censorship is that it has a very unique way of handling the root zone which I will go into more detail later. So this is very high level, what DNS is, what its motivation is, and what its features are.

Now, in the context of ICANN, what is probably more even more interesting than the security properties are how zone delegation and the root zone actually works. So in GNS, the NS record equivalent is called PKEY record. PKEY record for those familiar with DNSSEC is actually quite similar to a DS record or maybe some kind of a mixture between an NS and the DS record. It simply contains its own public key.

So in order to delegate authority over a name to another zone you simply publish a PKEY record which contains the identifier for the zone that it's delegated to. It's easy to understand this, for an example. So let's say Bob registered a subdomain at the .com zone and now he wants to publish the IP address to his web server.

So eventually what's supposed to happen is that www.bob.com should point to 1.2.3.4. Now as .com agreed in this example to give Bob the bop.com subdomain, what it will do is it will publish a PKEY delegation record into the distributed directory. The value of the PKEY record will contain the public key of Bob's zone which he is administering.

Bob can generate this public and private key pair locally as long as he is registering it with the .com administrator. Bob in turn can publish the IP address he wants to be reachable under www in his own zone in the directory. What happens is that for the PKEY record, Bob and the zone key 5G0Z in that example, is used to derive a DHT key that is then used to store the record under and is also used to decrypt the record. And the same happens with Bob's www record.

How would that record be resolved? Let's say Alice wants to resolve www.bob.com, what she will do is she will first derive a key from Bob and the root zone public key, 5G0Z. That will [inaudible] with the PKEY record to Bob's zone. When she has that record, she can recursively resolve www in the zone she has just learned from, which will then yield the IP address.

Now what I kind of skipped over is how is the top level zone bootstrapped? Specifically, how did, how did Alice in this case learn from .com, which is probably most interesting here. There must be some way that Alice actually knows that .com points to 5G0Z or else she cannot make the initial query for the subdomain. What GNS currently implements is something called a hyper-hyper local root concept, which means that each resolver ships with an initial root zone configuration.

Now by design, this initial root zone configuration is configurable locally at each endpoint. This is important, or else we cannot realize our censorship resistance, because if some subdomain were to be censored, the user must have the ability to basically skip a delegation and then still be able to resolve the actual record.

So you think, if somewhere in the world Facebook.com is censored, then you might just as a user put into your local root zone a delegation from Facebook.com to the actual authority that is responsible for that zone. So this override or this extension of the root, if you want to call it that, is possible not only on the top-level domain level, but also for

any subdomain selectively, in order to circumvent any kind of censorship that may be happening.

But it is also possible to use this to realize private networks. So, we have also heard about the issue some companies have with DOT and DOH, that it is impossible to realize private company networks, using that system. In GNS this is still possible.

You can simply have all of your employee devices shipped with a specific root zone the includes your private network delegations and your private network top level domains in order for them to retrieve internal information on internal services. But at the same time those records are still protected and the queries and responses cannot be tracked.

But the thing is, how is this initial root governed? How do the developers decide what top level domains should be put into the default root zone? Obviously the issue here is that there must be a governing body that defines what should be put in the default root zone. And here we have a very innovative and original idea. We wanted to create a nonprofit organization with a multistakeholder model, including a Board with the developers on it.

And then we're going to look for supporting organizations such as free software and operating system distributors, browser vendors, maybe even governments, if they want to have EIDs in GNS. And then for operation expenses we could simply auction off new top level domains for brands or companies.

So, where is GNS and where is it going? At the moment we have received funding from internet under a grant from the European Union, Next Generation Internet, in order to create a technical specification for the protocol. So we currently have a reference implementation. But as there should be more implementations than one, we are currently in the process of creating a specification packaging and alternative implementations. Also, we are continuously developing and integrating it into applications.

Now, as I already mentioned in the beginning, there are already quite a few applications using GNS so we more or less need a governing body for our initially shipped root zone, so we are planning to do that in the near future. That's it for me. Thank you.

ADIEL AKPLOGAN: Thank you very much, Martin. Good, we will now move to the second emerging technology that we'll talk about, this is Handshake. Boyma Fahnbulleh will present that. Boyma is the Vice President of protocol engineering at Purse, which is an e-commerce company, but that contributes a lot to Handshake. So we'll hear from you.

BOYMA FAHNBULLEH: Yeah, thank you Adiel. So yeah, I'm Boyma and I'm at Purse. We contribute to the Bitcoin protocol as well as Handshake protocol. Some things I've worked on as a contributor here has been embedded software systems for hardware wallets and even like working on the marketing side. So I've seen like everything across the stack.

So Handshake, we'd like to call it an experimental P2P root DNS and essentially what that means is that the protocol is seeking to solve Zooko's trilemma, or Zooko's triangle as some may call it, which is essentially the idea that within three certain aspects of decentralization security and human readability between a network naming system, you can only choose two. And we think that we can actually solve this using UTXO based blockchain.

At a high level, you could think of a blockchain as essentially a distributed database system that instead of replicating itself across different nodes each node seems to hold one source of truth that is shared across the network and the UTXO is essentially the mechanism that updates the state of this block chain.

So what we've looked to do is actually tie name ownership to the ownership of one of these UTXOs, and by doing so and relying on the proof of work of the chain, you can create a chain of trust based off of a digital signature tied to the key that owns that UTXO. And then the decentralized network of validating peers on this network would act as a trust anchor for this root zone.

So, as I said, we are blockchain based naming system and a lot of questions I get when speaking to different groups, especially at different blockchain conferences, is Namecoin exists, Blockstack exists, ENS exists, why does Handshake need to exist?

There are a couple different reasons that we find ourselves necessary and also different from the aforementioned projects. One has to do

with the fact that all of these projects tend to take the approach of, I don't know that they actually register special domain names, because it's kind of a hard process, but the idea is that they bifurcate their namespace underneath some sort of specialty TLD and then administer names under that as subdomains.

What we're actually proposing to do is actually creating an alternative root zone. So the chain would be managing and issuing new TLDs as opposed to subdomains under the current root. We also look to be completely compatible with DNS because again we're not really trying to replace DNS, it's just replacing the root zone and the root servers with a decentralized network of peers.

And actually Namecoin would be a bit harder because Namecoin itself is its own decentralized blockchain with its own miners and security mechanisms but Blockstack and ENS for sure could definitely be projects built on top of Handshake and we've actually facilitated a path towards that, we will see in the future if we can collaborate with these projects.

Namecoin could as well, but based on the fact that ownership of the UTXOs is based off of some sort of key, either a single key or a quorum, I don't know that their system maps from a decentralized network of peers to a three or five multi SIG or something of that nature.

So, I just want to give a warning, I'm going to give a really basic explanation of how DNS resolution works only to build some intuition about what it is we're trying to do. So, forgetting about the OS

number resolver, we're assuming just a client is making a request to the DNS resolver on the right, we can imagine the DNS resolver is like, yo, where is google.com? And it's going to make a request to a root name server up at the top right.

This root name server won't know where google.com is but it will know the whereabouts of the name server over the com domain. So it says, I don't know, but you can ask this name server and maybe they'll be able to help, we'll iterate through, hey, where's google.com to the com name server. It'll answer back, I don't know, but ask this server, and eventually we hit the google.com or, you know, probably ns1.google.com authoritative name server over the google.com name and it will reply back with a record saying, hey, I know where the IP address is.

Now to reiterate what I was explaining on the last slide, we are actually looking to replace all of the root name servers with our full nodes. So on Handshake, when you buy a name, you're not buying a subdomain from a com or a net or an org, you would actually be creating your own TLD.

And again, as I said, we're doing this because the hope is to solve Zooko's triangle and to get a little bit more into the idea of Zooko's trilemma, as I stated earlier, it's basically a decision game between two of these three properties, and you can imagine, I like to think of decentralization or the decentralized aspect as the system being able to work trust boundaries, let's say. Peers can connect with each other

and they don't need to worry about vulnerabilities between each other or a third party.

Security is about name resolution and being sure that you are resolving the name to the correct server that has the data you're trying to download and human meaningfulness is about making sure that names can be understood and also memorized by people because we are who use the internet. And I would venture to say that DNS currently solves for security and human meaningfulness, whereas with Handshake, we think we can get all three.

So another piece about Handshake, and actually, I would say, and I think some of the other project creators would say that this is the bigger piece of what we're trying to solve, is a decentralized certificate authority. We're all familiar with CA's and the centralizing, I guess just like vector of trust that we place in them as an internet society and these organizations can be hacked and when they're hacked, certificates can be issued and things get nasty on the internet.

And so we think that if names are again actually rooted to a blockchain then you essentially have, you can imagine the Handshake blockchain almost it's like just a decentralized data availability layer so you can peg things to the blockchain and store it in the state. And now, as long as you trust that this blockchain is the source of truth of this network, then you can trust the records at the root of this block chain and then create a chain of trust down below that, which don't even have to store their data on the blockchain.

We believe it scales actually quite better than some of your typical blockchain projects because the amount of data that needs to be stored on the chain doesn't balloon out because most of the subdomains, because of the hierarchical nature of DNS can rely on sublayers of the chain.

And this can be solved many ways. We could use TLSA records stored directly on chain. You can imagine a system where public resolvers off of this network run either full nodes or like clients who are using full nodes to validate the state of the chain and then allowing them to trust all records within the chain and then build that trust downstream. This is a current area of research, but there's a lot of different ways you can actually accomplish this.

Okay, so we talked about the fact that this chain essentially is going to be issuing names, but we don't know how, I haven't explained what's actually going to be going on. And name registration is actually modeled as a victory auction or a second price seal bid auction.

And you can imagine a second price seal bid auction having the same price discovery mechanisms is like your typical English auction, like if I'm the auctioneer, I ask who wants to bid a dollar for this name, you raise your hand, and then I go who wants to bid two dollars for this name, someone raises their hand, and then we get to three dollars presumably, and no one wants to bid three dollars, so this third bit has actually kind of been seen to the group of bidders, but the second price actually wins, because no one actually wants to bid this third or

the ultimate price. And so essentially we've modeled this as a state machine that exists on the coins.

So, I talked earlier about the UTXO and you can essentially, at a high level for people to understand, transactions on the Bitcoin network as well as our Handshake network involve, it's similar to cash. It involves spending outputs that were created in previous transactions. So let's say I have an output that's worth 5 HNS and I want to send To 2 HNS to Martin.

I would construct a transaction that has the state of the inputs of this transaction as my 5 HNS output and then I would create two new outputs one 2 HNS output that spins to Martin and pays him and then a second output that comes back to me as change. Typically on these outputs you have an address, which is essentially an identifier for who is able to spend this coin and also an output value which is the amount that you're trying to spend.

With Handshake, we've added a third consensus object which is a covenant, and this covenant has a type and then also data that exists on this covenant that helps to facilitate the auctions. So you can imagine, and I'll get through this a little bit more through the slides, but these are some of the covenant types that we have. So we have an opening covenant. And so you can imagine I have some coin somewhere in a wallet and I decided I want to open up an auction for .boyma, which is my name.

So I would actually create a transaction that spins my unencumbered coins into a covenant that is an open covenant and what this covenant says, you can imagine a covenant is just like a promise, in the same way that you think of the traditional English word covenant, but this covenant is a promise that the next time this coin is spent, I can only spend this coin in a bid transaction which will actually bid on the name that I specified in the open.

Once it goes into the big transaction, that bid transaction is a covenant that says I can only spend this coin if I want to then reveal the actual price of my bid. And I guess I left out the fact that the bids are all blind. So if say you're bidding 5 HNS you would actually use 10 HNS to obfuscate the price, the cost of your bid and then at the reveal stage, you would then reveal that price of the bid. And so you can imagine all of our different covenant types exist as a state machine that model these auctions and these auctions are how the names are actually issued on chain.

Now name management also exists in this covenant state machine manner. Once you have actually won an auction, you would spend into what is called a register covenant or update is logically the same covenant type and that allows you to actually store the name records that you would like to tie to this name in the embedded state of UTXO as well.

These names expire every year, so you would need to renew every year. Renewals are free, except for the cost of a transaction on the network. And you can also transfer these names to different users on

the network and also revoke the name. The reason you would want to revoke the name is because these names are actually tied to a cryptographic key.

If your key was to get hacked, you would potentially maybe want to revoke the name as like a economic game that disincentivizes someone to try to steal the name, because if you steal the name, I can revoke the name from you but if you have questions, I can answer that later. It's kind of like a high level overview for this time.

I won't get too deep into the weeds on the Urkel Tree, but the Urkel Tree is essentially a commitment to the state of all names in the root zone. We implement is as a base-2 merklelized trie where you can imagine a tree like structure where the keys to the database form the path down to the leaf of the tree, and the leaf of the tree is actually the name state or the records that are stored chain.

One caveat to that is that we don't actually store records serialized as DNS records on chain. We actually have the idea of a resource and this resource can be deserialized as DNS records on chain. We do have for a couple reasons that I can get into later. We actually have two variants of the tree. Essentially this is to mitigate bit collisions on block headers. For the sake of time, I won't get into this too much but I can come back to this if someone cares about authenticated data structures.

So, we talked about how the names will actually be issued on chain but one issue that we have to think about is squatting, not only

squatting, but also if this is going to be an alternative root zone system, what happens to all the TLDS that already exist in the current root zone.

So what we've actually done is we've reserved all known gTLDs at present in the chain and also we went through the Alexa Top 100,000 Domain Names list and reserved with three duplications 80,000 names as TLDs, as well. And these names can actually be claimed using DNSSEC proofs. So we hope that this helps with adoption of DNSSEC, because I know right now it's very not widely deployed.

And so there's an economic incentive assuming, the Handshake chain gains any real world traction or value to claim these names and to claim these names, you need to implement DNSSEC proof. So hopefully we can get some good percentage of the top 100,000 most popular domain names on the internet to actually care about security.

Okay, I was going to explain DNSSEC, but for the sake of time maybe we can skip this and given the audience, I'm sure you're all familiar with DNSSEC. So we actually have an airdrop that we are about to administer which our idea around this project was to incentivize the traditional internet community. There was a lot of work done to create the internet as we know it exists and a lot of that work was done through the goodwill and enthusiasm of the old school internet community.

And so with a lot of blockchain projects, I don't know, I don't want to get normative, but they seem to be money grabs and we thought, how can we flip that economic game.

So we actually are doing like an airdrop that's going to distribute 70% of the initial coin supply to open source developers and the cool thing about this airdrop is that it is privacy enhanced so we looked at the PTP web trust, all get up users to have SSH keys and then any Hacker News users that are also on the key base PKI system, and we've actually left claims on coins for them on our chain as well.

And these claims can be redeemed through signatures using your PTP web trust key or one of your SSH keys or a Keybase key. And that key, though, or that signature is actually not a proof of knowledge of the private key that's tied to whatever identity it is that you have. We've actually generated a scaler for any one of Ed25519 or secp256k1 curves and we modify that scaler with your key which allows you to create a proof on the modified key, not your key, so you're not actually losing any privacy. You're not giving up your identity.

Now there is a set of users that could get this airdrop and you are a part of this set of users, but when you redeem there's no way to know that you're the person that's redeeming. I guess that's it for like a high level overview. But again, we're here to answer any questions. So, I'll just talk about the software we have right now.

There's a full node which acts as a root name server. You can run it as a recursive server as well. But we imagine that, like any full node

operator is existing as a root name server for the root zone, and it's written in JavaScript. And then we have HMSD which is a light client, and we hope it will be used as the recursive resolver, which can make proof requests to a full node and validate the state of the root zone or at least the key value pair of the data in the tree.

So, if you have any questions, handshake.org or you can check the source code at github.com/handshake.org. We have a community website Handshake.community.

But as we grow the community, I would be remiss not to say that we are definitely more of a group that's involved in the block chain space and we are welcoming and hoping that more traditional internet and DNS in particular minded people contribute to our conversations and even review the code and bring opinions, because a lot of our problems deal with decoupling our protocol layer, which in a sense, has nothing to do with DNS, to the application layer that is DNS on top of that protocol and I can imagine in the future while right now this initial project is about creating alternative root zone to create a decentralized certificate authority at the end of the day, what we have is a decentralized data availability platform and you can store whatever type of data you want in this chain and we're open to ideas about what type of alternative naming systems that could exist and what this could create.

So, that's it for me. Feel free to email me and I can, try to put you in contact with the parts of the community that are more in tune with

what you're trying to talk about, and I'm willing to do that for sure. Thank you.

ADIEL AKPLOGAN: Thank you very much. Interesting presentation and explanation as well on what Handshake is bringing on the table. So, our next presentation will be from Paul. Paul is principle technologist at ICANN and works within the office of the CTO Mostly on identifier system standard and research relevant to that. So he will give us a little bit of a view of the DNS today, the different experience we are seeing around its evolution. So, Paul?

PAUL HOFFMAN: So, as Adiel said earlier, this is the fourth time we've had emerging identifiers and so we've had six or seven emerging identifiers over the last three years since we've been having these presentations. And there is really the question, is there an overarching picture of what is an emerging identifier? So that's what I'm doing today.

I'm not bringing you a new emerging identifier. I'm trying to set the view that you might have because some of the emerging identifiers, we've heard today and previously, may turn into something big later, many of them will not, just as many of our ideas often don't turn into anything. But it's not like they're all competing. They in fact, many of them are cooperative. Some of them are competing, but it's hard to tell if they are so what I intend to do today is to sort of give you an idea of how to tell what is this whole ecosystem.

So as you look at emerging identifiers, or you see one come by as you're reading some news articles, you might have an idea of what is this, and how does this relate to the DNS. So I'm going to start with a little bit of vocabulary because vocabulary is important. Many of the things you heard from the first two speakers today were similar, but they were using different vocabulary, which makes total sense because groups of people, once they start getting excited and working on a particular topic, will sort of gravitate towards their own vocabulary.

And it's nice to have sort of an overview of it. It's not like everyone should adopt the same vocabulary, but at least we have some examples here. I'll go into a little bit of history and I really do mean a little bit, because there have been plenty of emerging identifiers over the last 25 years that could be considered in the namespace. And then I will have a couple slides on how to differentiate an emerging identifier from the DNS, or how to make it look part of the DNS, because that's a real important thing to people here.

I know many people in this room didn't understand any of the technology from the first two speakers, but in fact may care about what's the result of this. In the same way that many people in this room who haven't taken Our How It Works DNS Introduction that we offer twice every time, still are here and are very concerned about how does this relate to names.

You don't have to know the technology in order to do this. There are plenty of people who come to ICANN meetings who don't really

understand the first thing about the DNS. I wish you would come to our How It Works stuff, but even if you don't, you care about names. So that's what I want to go into here.

So let me start with definitions. Definitions are important. The IETF created an RFC called RFC8499 called DNS Terminology. Very basic, but it has pretty much, if you care about the DNS and you want to use the terminology, it's pretty much got everything in there.

It's not a good introduction, because in the same way, I don't know, probably some of you are nerdy like I was when you were a kid, and you started reading the encyclopedia, and you started hopping around and saying, I want to know everything about the words that start with er, or whatever, but doing that's not a really good way to learn about the world. But, you can use RFC8499 as a way of finding out the parts of the DNS where you do know the name.

The one useful term that I'm going to bring up here, which is really good, is global DNS, because many of you may have had this problem, you're talking to friends who aren't in the technical field and not in our field and they say, well, what is the DNS? And then you just start babbling a little bit. So the global DNS is a naming system.

And in RFC8499 there is a definition of what is a naming system, not just the DNS, but what are the important aspects of all naming systems. Because we all inherently know something about naming systems. We are used to calling our friends by their names. We are used to the fact that we might have two friends who have the same

name, so we sort of get that. So a naming system has a bunch of facets. Composition names. What goes into a name. What is the format of them. How do you see them, how do you hear them, things like that. How are they administered?

In humans, your parents get to give it to you, except if the government says no, that's not an acceptable name. Types of data that can be associated with names. Both the people earlier today were talking about the wide variety of kinds of data that can be associated with the names and their name systems. Types of metadata, which is a little bit off track. The protocol for getting the names.

If I meet you for the first time on the street I would have to ask you your name. If I'm here in the building. I would probably look at your badge. So that's two different protocols, one is audio and one is visual, but also it's from different sources. If I asked you, you're pretty authoritative for who is your name.

If I look at your badge, I'm trusting that whoever made the badges at ICANN spelled your name right, which isn't completely perfect for those of you have been looking around at the badges. So that's the protocol for actually getting a name. And then the context for resolving a name, now that I know your name can I yell it to you across the street? Do I have to type it in a certain way, things like that.

So these are the general, if you have a naming system, these are things that might be interesting. So now let's look at the global DNS and again these definitions come from RFC8499. So the composition

names in the global DNS, it's one or more labels, the length of labels, there's a couple of other technical things. That's how a name in the global DNS is formed.

The format, there happens to be three of them, just to make things a little bit less interesting. There's the wire format, the way it works in the protocol. There's the presentation format, the way that you might see it in an application, which isn't always exactly the same way as a common display.

When you walk out in the hall here you see the advertisements up there, you'll notice that people display their domain names in different fashions to sort of show things out. So those are all different and we're all used to it. We in ICANN sort of have gotten used to this.

The administration of names. How do names come out? It's by delegation in the global DNS. There's the root that everyone knows, that delegates to the TLDs, each TLD delegates to a second level, and so on. The types of data, could have no resource records or it could have a lot. The kinds of data are addresses and text records and keys and things like that. So we're all sort of familiar with that even though if we only really know addresses, we believe that there are all these other ones.

The protocol for getting names, RFC1035, one of the very early RFCs, 30 years ago, it's pretty much still the way that everyone gets their domain names. There are some new transports, but basically the protocol has been around since before almost any of us, not that we

were born, but since almost any of us were starting to use the thing that became called the internet. And the context is the global DNS root zone distributed by IANA, that is, these are hierarchical names that are all rooted in IANA's root, that's the context.

If you ask somebody in a different context about a name, you might get different names, but you need to have a context when you ask names, just in the same way that if you walk into a party and say, is Bob here, and you walk to the party next door and say, is Bob here, the answer might be yes in both places, but because the context is different you will actually get different answers.

So, let's talk a bit about emerging identifiers. The types of identifiers that are managed by IANA, the things that we're familiar with here, are names, addresses and then the very large things that everyone calls the technical identifiers. So you could have emerging identifiers in any of those three categories, but the ones that we've seen the most are names. That's really what are attractive to people.

Think of addresses, which are mostly for hosts of course on the internet. But as like your house address. What are you more interested in your name or your house address? It's your name, you want people to pronounce it correctly, you want them to remember it.

So that's why the emerging identifiers that we've seen over time are often around names. There's a little bit in addresses, but we don't find that much, in fact, we have not had an emerging identifier session on

addresses, because that usually just goes way over everybody's head, even though there are some possibly interesting work there.

So now that we're talking about emerging identifiers of names, the earliest ones, and it's the easiest one to do is just replace the root. You have a set of names, I have a set of names. Why don't you follow my set of names. And there were two ways that people did that, they essentially just created a completely new root.

With non IANA TLDs and if you asked for an IANA based TLD, it would say I don't know anything about that, go ask your normal folks. But the other, is that reproduce the IANA TLDs and then they add ones that so far have not been. And of course, there's a failure when IANA later adds that.

Collisions are always a problem. They're a problem even if you walk into a party and say, is Bob here, because two people might say, well, yes, I am. Or one person might say no, Bob is not here, and one person might say, Yes, I am. Because that first person didn't know that Bob was here. So collisions and anti-collisions can always happen.

So none of these early ones from 90s that the early 2000s really caught on. Some of them still exist. So it's not, again, you have an emerging identifier for a purpose, you can keep it running forever. Do you get a lot of traction or not, I don't know. And so these never caught on much. But let's talk about some of the other emerging namespace and some of which we've actually talked here. The ITU actually has an

emerging identifier namespace, which is ASN1 which uses UTF8 as labels instead of integers.

For those of you who are familiar with something oids, it's integer dot integer dot integer, well you can actually use UTF strings in there starting with ITU TX690, which is about five or six years old. So that is an emerging namespace, it hasn't been used much, but a lot of people here know who the ITU is, so it could be. DNS-like names that are actually in the global DNS we've seen some of those, [inaudible] which had a talk on a few years ago.

And if you think about it, anyone who's doing a creative use of the DNS themselves just even within their own organization where they're using names in a way that people looking at the name would recognize it. That's an emerging identifier. It's very local. It's globally scoped. Anyone can sort of see into it.

But there's lots of reasons do that and some people can get really creative. I mean we at ICANN when people say, hey, we're doing this. I'll just look at them and go, whoa, that's weird. But it works for them. That's all that matters. It works for them. Another namespace is the TOR namespace with dot.onion, they got the .onion TLD reserved as a technical top level. And they're doing whatever they want underneath with that.

And then there's, as we've heard, there's some of the blockchain based name systems like the Ethereum Name System or ENS, but there are many of those, and when I say many, there really are many.

We're not going to have a session for each one of them because some of them come and go very quickly, some of the merge with each other, and actually, as we heard from both speakers today, sometimes they actually merge technically, that is that you'll have two of them running sort of in parallel, but they could cross over. Again, emerging doesn't mean it has to be global.

So this is my either last or next to last slide. Sometimes these names look a lot like DNS names. And so if you want to look like a DNS name you just use the global DNS presentation format like www.example.com that's something that everybody, people who barely understand the internet actually understand the format of a domain name, which is sort of scary. A lot of these people don't even know really how to run their browser, but they would say that's a domain name and that's not.

Visually names are super, super important. It even is important verbally. When you say I got that from, and you just say one label, like Facebook, that might be a place or whatever. But if you say I got that from facebook.com people recognize that as a domain name. So the presentation format that people invented over 30 years ago not only stuck, but it is universal in people's heads at this point.

So, if you create an emerging Identifier that has that kind and you want it to stay local in your naming system, it ain't gonna happen, because people think, oh, that's something I can look up on the internet. They don't know that the global DNS is or is not the internet. So those kinds of things will always leak out and we've seen plenty

examples of that, where people said we're going to have this naming system that looks like a DNS name, but we're going to keep it local. And we see those going out on the internet, all the time.

If you run a root server, if you run even a resolver with a couple of people behind it, you will see those names, coming out, that's called leakage. Or you can choose not to look like DNS names. So you can use a similar presentation format, but just use something in between instead use colons, www:example:com or backslash if you really want to be nerdy.

When people see those, they don't think that it's a domain name they understand that it might be in a different context. They don't know the context, colon doesn't tell you the context, but let's face it, a period doesn't say anything like this is the internet. It's just a period. They're at the end of every sentence, and yet people recognize those.

So if you want to not look like that, you use a different separate if you're doing a hierarchal name. The advantage of that is those things, if you're trying to keep them private, is those are much less likely to appear in the global DNS. And so that keeps it sort of in the area that you want, in your application, in your network.

So this is my last slide. Summary is that identifiers will emerge. They've been emerging for 30 years because some people didn't believe that the original DNS was such a good idea, and they came up with their own, anyways, and that's great. In the same way that you, can put up whatever you want, as long as you can get a website, you

can put up whatever content you want, content is one thing, naming is another. There is no reason for anybody to say you can't name things that way.

Now they might say you can't confusingly name things that way to not fraudulently make it look like something and unfortunately, the DNS allows that completely because all names are unique. But there's no reason to say stop coming out with emerging identifiers. That's just silly because people are creative and people have local uses. Remember, we've only had internationalized domain names for 15 years.

Before that, what we were inherently saying to people was, you can have any name you want as long as you spell it like Americans do. And let's face it, the internet was quite international well before that and it took that long for people to go like, oh yeah, okay, what are we going to do about that. And if you're at all into IDNs by all means go to the universal acceptance, the UA table, outside, they're still fighting with this. So, even on that level, just saying, you can have any name you want as long as you spell it the way you want, that's a limitation.

So emerging identifiers are going to happen. Sometimes they happen and they succeed, and they grow. They often grow in ways that the originators didn't intend. Much of what you see in the DNS today is not what the originators intended, but they knew that this kind of thing would happen. So they put in a lot of expandability and many of these emerging identifiers can just work in the DNS. You don't have to create something new and wild and do your own thing.

You can actually do it in the DNS already, or if that doesn't work for you, for example, especially the GNU naming system, they had reasons why they didn't want to do it in the DNS. Create your own, that's all fine. So, this was a bit of history a little bit of future and I'm out of time, so we're ready for questions.

ADIEL AKPLOGAN: Thank you Paul, for reminding us of history and also the importance of the naming. So, we'll now move to the second part, which is interaction questions with the panelists. So people who have questions on the two presentations, the GNU System and also Handshake.

JIM GALVIN: Jim Galvin for Afilias, for the record. I actually don't have a question, but I did want to come back since we seem to be a little calm here and point out to Paul, I very much appreciated the presentation, I thought it was a great presentation. And actually, I think the notion of putting emerging identifiers in context, I think it does a very good job of that.

And I'd like to see that presentation appear more often and in more places. I mean seriously, you smile, but I think every time we have one of these sessions, some form of that presentation, I don't know if it's possible to make it a little shorter or maybe just what it is, some documentation, I think would be an excellent thing.

I think it's very good to remind us all why we're here, why we examine these technologies, what they mean, from my point of view, just putting everything else that we're doing here in context is very helpful. And I thought your presentation, did that. So, thank you.

ADIEL AKPLOGAN: Thank you, Paul. Thank you Jim for that comment. Good.

YOSHIRO YONEYA: This is Yoshiro Yoneya from JPRS. My question is to Martin. During your explanation regarding the bootstrap you used facebook.com example for alternative to the original bootstrap instead of the user's replacement. So I think it can create alternative tree, or it breaks the uniqueness of the name. So I was very confused if people can use it as their original bootstrap file, the uniqueness of the name will be broken. So that is my question.

MARTIN SCHANZENBACH: So, general, yes, that is the idea that should be possible to essentially locally for you to break this uniqueness and to basically step out of the original hierarchy and say, okay, I need to, for example, circumvent this because, because for example, if you have access to the whole .com namespace through the .com zone but for some reason all of the names in .com are available for you through DNS except for facebook.com, what you can do is you can use DNS and point it directly locally to the authoritative zone that is managed by Facebook,

for example, in order to circumvent the censorship of this one single name.

And yes, what you could do locally is you could point it somewhere else, so you could point somewhere where it's not actually Facebook, then locally for you, this would point to somebody completely different. But that's fine. That's just the concept of the local root. I think there's even an RFC for that, how you can actually manage a root server locally. So this idea is not so new, but yes, it's exactly like you say, the uniqueness of the name is then broken but only locally, because this is not published for anybody else.

ADIEL AKPLOGAN: Yeah but that defeats the purpose, because you want to circumvent something and then you are kind of creating a duplication now, which at the end I can manipulate that as well, to send you to the wrong information. So you are not actually circumventing at the end of the day, but it's opening the door for redirecting you to the wrong name.

MARTIN SCHANZENBACH: Yes, but who would redirect me to the wrong name?

ADIEL AKPLOGAN: The person manipulating the data locally.

MARTIN SCHANZENBACH: Ah, so if you are exposed locally, for example, through a virus or whatever, I think you have bigger problems than naming.

PAUL HOFFMAN: Let me hop in here on two things. One is that I think people might, and Adiel, it was important to understand that what the GNU naming system and others, in fact, I think this is probably also true for anyone who's also actually running Handshake at the time, names are no longer unique, necessarily, but they are completely controlled by the user.

So you are not going to get fooled by somebody else. You can fool yourself, we do that all the time, but it's up to you. And what this does to some extent is it takes the same idea of you might run this browser or this browser, you might get different views of the same thing, and that takes it all the way down into the domain name system. So the global DNS that can't happen but some of these emerging identifiers in fact say you want to control the way that you resolve names differently and maybe fool yourself.

Now it's not fooling yourself if you're actually doing this for circumventing censorship. It's not fooling yourself if you are willing to have stale data because you're worried that the newer data is wrong in the global DNS or whatever. But more importantly, when you said, Oh, well, there's a way of bringing down the root zone and doing that for local changes; absolutely not.

And I say this as author of the document, RFC7706. All we describe is how to bring down the root zone so that you can do local resolution without talking to the root servers, but because you have to be doing DNSSEC validation, you will never actually lie to yourself. So that whole effort is to do a few things. It gives you privacy, which is one of the things you all are concerned with because now your queries to the root aren't going anywhere, they're staying local for you.

But the other is it gives you speed. Many, many of the queries that things are asking for are for things that don't exist. So, why sit around and wait for, you know, even if it's 10 milliseconds or whatever, for something that says nope, that doesn't exist, might as well know that locally. So what we have currently in the DNS does allow local root resolution, but of the actual root contents without modifications. Does that make sense?

MARTIN SCHANZENBACH: That is as long as I use the correct DNSSEC trust anchor, I guess.

PAUL HOFFMAN: Correct, or even if you're not doing DNSSEC, as long as you get the root zone from an authoritative source.

ADIEL AKPLOGAN: You want to add something?

UNKNOWN SPEAKER: I guess the one thing that I would add is our system is more akin to what Paul was describing. I guess the difference is that at some point in time, we're anticipating a bifurcation of our root zone from the IANA led root zone simply because we can't assume that people won't buy, if you let people buy whatever name they want, they're going to buy names that people will probably want to gain through the traditional system.

And once that happens, there are quite a few ways that we can mitigate this split brain situation that will arise, one being, there's been talk about the community sort of choosing to always fall back to IANA within some sort of conflict and name resolution. I think really what will happen is we'll see a market form for different public resolvers that maybe stick to the Handshake root versus falling back to the IANA root.

PAUL HOFFMAN: And just to be clear, I think that will also happen under GNS, as well, that even though they're all going to start using the current Distributed Hash Table, at some point where there might be conflict there will be like you just said, some resolvers will be known to do it this way, and some will be known, and that's true pretty much for every emerging identifier that looks like the DNS.

UNKNOWN SPEAKER: We've also talked about having some sort of like meta name which completely mirrors IANA in perpetuity, and then it's just like one more

subdomain on top of resolving whatever it is you're trying to resolve. Said a different way, you can imagine the meta zone of _HNS and then so if you want to go to facebook.com you could go to facebook.com assuming there wasn't a change, or you could go to facebook.com._HNS and know that will always resolve to the IANA root zone.

ADIEL AKPLOGAN: I just want to prevent Paul and you to hijack the whole session. So there are a few other questions. Wendy, thank you, you are in the queue.

WENDY SELTZER: Thanks, Wendy Seltzer. I also found the presentations very interesting and I wanted to highlight a point that you made, Paul, where you pointed out, the multitude of options available for naming and the period tends to signify to people that it's in the DNS and not an alternative name system. I wondered, is that an opportunity for academic research, is there research in that area and would it be interesting for more data on that particular perception?

PAUL HOFFMAN: I haven't seen any, but I'm not an academic and you work at an academic institution, Wendy. In fact, you work at the academic institution that I went to as an undergraduate. So, I would hope so. I don't know. You all had to sort of make decisions at one point about

what were you going to use in your names. Did you all do any research or did you see any others?

MARTIN SCHANZENBACH: So, for us, the reason why our names look exactly like DNS names is because people are used to those names. And in order to have proper usability and to not alienate people in any way, that was the way to go. So I don't know this is the case, but I think that's just because it's been around for such a long time and there's not really a good reason to change it because that will instantly cause a variety of usability issues that we would have and would also reduce acceptance eventually.

So maybe one thing to add, so that's why I think usually what is the way to go for .onion, for example, is to use a special domain name. Because a special top level domain for technical systems to do that, but it's definitely probably not a good idea to simply change the naming scheme because it will be very difficult to ensure people to make people accept that.

BOYMA FAHNBULLEH: I guess I'll add, I spoke to this a little bit earlier in my presentation, but obviously, we chose to use DNS records because the goal of this particular use case of this block chain is to create an alternative DNS root zone. But, we actually store arbitrary data on chain and this data is deserialized in the form of DNS records. So you could imagine creating a secondary application using, we have a version bit on our

client, so like taking the diversion bit and then creating some other record serialization format that is completely separate from DNS and using this as our system.

I guess I speak for myself, but I definitely think that for our purposes, we want to see what other naming systems can exist when there is a root of trust that is decentralized and maybe Handshake exists in the future as some alternative naming system altogether that just allows you to fall back on the IANA root zone without having to change a resolver. In a world where we play nice maybe that's what happens. I don't know, something to think about.

ADIEL AKPLOGAN: Maybe research to look into and to have some more definitive answer. So there is a question in the back.

JAMES GANNON: Hi, James Gannon. No it actually pivots perfectly off of the conversation just now. So as somebody who came from the tech space and now lives in the policy and government space, I've followed emerging identifiers for a long time and a lot of them fall apart because they're built purely as tech projects and then when they try to scale, it doesn't work.

So the statement is that that's going to be your challenge, and then there's a challenge for the people in the room here, you can see even just from the conversation between yourselves and Paul, there are

people in the ICANN space who have been building this system at scale, at hyper scale for many, many years.

And often when, from my own personal experience, when we approach projects to try and help you out with the governance aspects of things, when you're looking to get to that piece where you're trying to scale, often the reception is not really open to it. It's, you know, we're a tech project we're focused so much on the tech that the governance is almost left to the side.

And if you really want to build projects that are going to really have the ability to scale out properly, it's the people in the room here that are going to help you do that. I'd encourage both yourselves and anybody else that's building in this space to reach out to the people in the ICANN community because you'll find that they're super open to talking about it and super open to helping.

ENS is probably the one that's done it successfully to date, I think, and that's shown in their scale. And I think anybody else is coming up into the same space, do come to ICANN meetings, do try and engage with us more. Most of us love working on this stuff. And the more you're able to engage I think the more you're going to build governance into your tech rather than building governance on top of the tech, which usually doesn't work very well. So it's great to see you guys here.

ADIEL AKPLOGAN: Thank you, James. Another question here? Sébastien?

SÉBASTIEN BACHOLLET: Thank you. Thank you for the presentation Paul, I think that your presentation could be very useful to be done in other part of ICANN, not in a small specific group, I guess in At-Large it could be useful that you come to give us your presentation. Because one of the points is that how we as end user, individual end user, we are sure that we go to the right name and we are sure what we get if we have different identifier.

And my second point is that it's raised the issue of the brother who want to take out the image of his domain name written in the search engine because it's part of the things that the end user need to see to be sure that they are going at the right place and I am very concerned with what is happening specifically on Google, if they don't show the domain name anymore. Thank you.

PAUL HOFFMAN: So, that's actually an interesting one. I don't know if you guys caught that. But the fact that Google in the search now is thinking of not showing the domain names, and that goes back to the importance of naming. Another story related to that because many people in 2003 considered IDNs to be an alternative naming system.

You know, even though we in the IETF who had worked on it, we made sure it worked just fine with the DNS, when people started seeing those names, there were like, that can't be a name, it's got an accent in it, or it's got Chinese characters, or characters that look Asian, I

don't know if they're Chinese or Japanese, or whatever. So people considered that an alternate naming system and some of the browsers, in fact, would not show the IDN in the location bar initially.

Similar to your concern about us not seeing real names in search results and such like that, as some of the alternative identifier systems come out, they need to think about how are their names going to be displayed or hidden. Some of them might want to hide them, especially on many of the DHT based systems where what's beneath the name, you have the human friendly name and then the 64-character key. Do you necessarily want to hide those?

Because people, even though they can't read the key, they can do a quick match visually if these two keys are the same or different. Hiding and showing is a real interesting one. And it's been a problem on the internet forever.

We had pseudonyms before we actually had real names for those of you are old enough to remember Compuserve, we had five digits comma three digit names and then we could associate those numbers with a real name, but it wasn't a name to the real number, it was a number to the real name, and some of you are like, I'm not going to show my name. So, thank you for the point Sébastien, it is really interesting about the names being shown and names being hidden.

YAZID AKANHO: Hello everyone, thank you for the presentation. My name is Yazid Akanho, I am coming from Benin, West Africa. I am ICANN66 Fellow.

My question is not directly related to the presentations, but I would like to ask it, because I think is the right place to ask. What is currently being discussed around Internet of Things, naming, and the impact of Internet of Things to the current DNS infrastructure. Thank you very much.

BOYMA FAHNBULLEH: I guess I should caveat that I'm not too, I don't want to say I don't about the Internet of Things because I will be a consumer of Internet of Things, I won't be on the bleeding edge of the research, but one thing that I will say is that with Internet of Things I don't readily see the need to push forward with research as it pertains to naming, or at least as it pertains to the human readable attribute of naming, especially because these are software agents that will be speaking to each other and they can speak integers, they don't need the human readable names for the for the communication piece between your fridge and the digital wallet that exists wherever you're ordering your milk from. That's my intuition about this, but I could be wrong.

MARTIN SCHANZENBACH: Maybe one thing regarding the Internet of Things, a lot of people will have their own devices which maybe they want to name them and that would eventually mean that they might need names that they might need to be accessed by third parties. And you cannot really give everybody in the world a free domain name for that.

So maybe you need to go back to kind of a pet name system where you have something like this is Bob's iPad but you can actually use a readable name to access Bob's iPad, because, you know that is your friend and that is your friend's iPad, for example. That's just an example, and in the IoT that might be more useful or required and then more useful than maybe a globally unique name but a name that I can use to access my surroundings, so that might be something that might be required in the future.

BOYMA FAHNBULLEH: Totally, to add to that, pen name system, it's a good paper. I guess maybe to explain a little bit more, if you're not familiar with pet names, you can imagine a pet name system is how your address book works in your phone. There is a phone number that exists, that's tied to a human being somewhere and when I get Paul's number, maybe I put in, what's your last name, Paul? Mr Hoffman, but Martin might put in Paul, or Adiel might put in Paul because he's more familiar with him, but that name still resolves to a phone number that is globally unique.

And so having a local naming system makes a little bit more sense, which I agree with you, then a global naming system that tries to issue names across all the different devices that may exist because these devices again can communicate. Once my phone calls your phone it's just using the number, it doesn't care if it's Mr. Hoffman or Paul, or Homeboy P, whatever you may go by.

PAUL HOFFMAN: So, let me hop in because I handed it off to you guys because one of the things that I think is very important for IoT is trustability that this is mine. This is not his, this is mine. And certainly any of the naming systems that tie into public key, which both of yours do, in fact, most of them do these days, allows me to sign the pet name, so I don't have to worry about that.

Now, normally this kind of localization isn't good. That's why we have a globally unique name system. But if you want to be local, you want to be sure that you're local and therefore either GNU name system or, I keep wanting to call you Handshake, not Handbrake, and some of the others like that may be useful for a segregated name system because you can actually get a higher level of security without having to go to the global Internet, it means you lose global but it does mean that you can have a stronger local. Does that help?

BOYMA FAHNBULLEH: Yeah, I've actually ideated on things similar to this. You could imagine, Paul, you may buy Hoffman, the name Hoffman on Handshake and then within the Hoffman, record store some master public key that we use to authenticate the different records that you store below in subdomains and then have a subdomain that is phone.hoffman which stores your phone number with an authentication signature and then people can authenticate the public

key on chain and that's just one idea, but you could imagine a bunch of different schemes in that way.

And in that way, it is sort of a pet name system rooted in a global name system because you can administer your subdomains as you wish, but it's still tied to this global root of trust on the blockchain and can still be authenticated with a chain of trust.

ADIEL AKPLOGAN: So all that was the forum of the scope of the namespace, does it evolve with the evolution of the technology in general? I think that a lot of work and research and evolution are going to appear as we move on. We are almost on top of the hour, I don't know if there is any other question, there is none that I can see.

I would first of all like to thank my panelists for the very good presentation and as we have noticed for all the sessions, always very likely, a lot of discussion, interesting aspect on all the emerging identifiers, and our goal has always been to continue to try to find a way to continue the discussion beyond the meeting and give people a platform to interact with people who come here to present what they are doing, but also to allow the ICANN community to look at what is happening next and how those things are evolving, particularly when from ICANN perspective, we don't dive into any of them to come back to the community.

I think we are going to continue working on that aspect of this session to make sure that we keep it alive. Thank you, Paul, very much, and I

think people want to see your presentation elsewhere. So you've got to get ready for that. Thank you very much everyone. See you for the next EIT session which will be at the next ICANN meeting.

[END OF TRANSCRIPTION]