

DNS Operator Role in domain management

“a proposed model to improve the DNSSEC provisioning”

ICANN53 Buenos Aires

Tech Day

Latour - June 22, 2015

Why? What's this all about?

- .ca has 104 signed delegations ☹️
- 11 .ca Registrars support DNSSEC (out of ~150)
- Registrars are not interested in DNSSEC
 - Provides no value add & is a DNS Operator function
 - It is a cost, every request digs in margins
- Provisioning model was designed around the Registrant, Registrar and Registry model (RRR)
- Need to redesign around the [DNS Operator](#)

History, Legacy & Sacred Cows

- 2004 - NLnetlabs suggested a new SECREG-C contact to handle DNSSEC material with direct access to registries but failed due to pressure from the RRR model to not have Registry talk to Registrant.
 - <https://nlnetlabs.nl/downloads/publications/dnssec/dnssecnl/secreg-report.pdf>
 - Problem just got postponed until now...

Food for thought

We have a **sacred cow**, conceptually, since the dawn of time, registrars have been granted full control of relaying & managing **ALL** registrant domain information to the registry.

No one stopped and asked when DNSSEC material was introduced if the registrar should manage this? Or **WHO** should manage this material? Or **IF** it should be the DNS Operator?

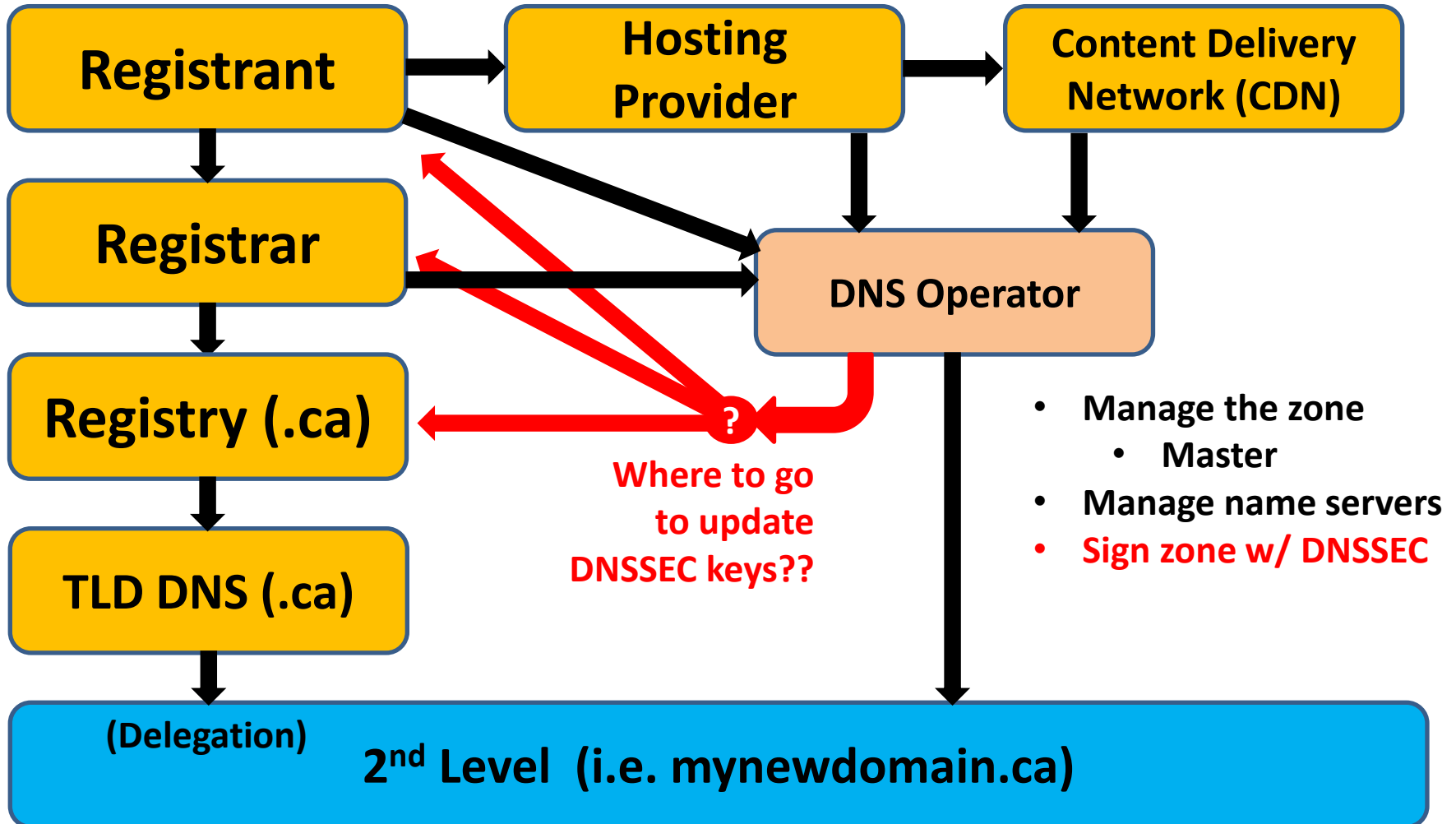
RFC5910 bingo, we all moved along and “assumed” it was the registrars responsibility and the registrars came back “hey!!! we don’t want to manage this *#%&”

Then we had DS or DNSKEY religion war, Key Relay, CDS, CSYNC, etc...

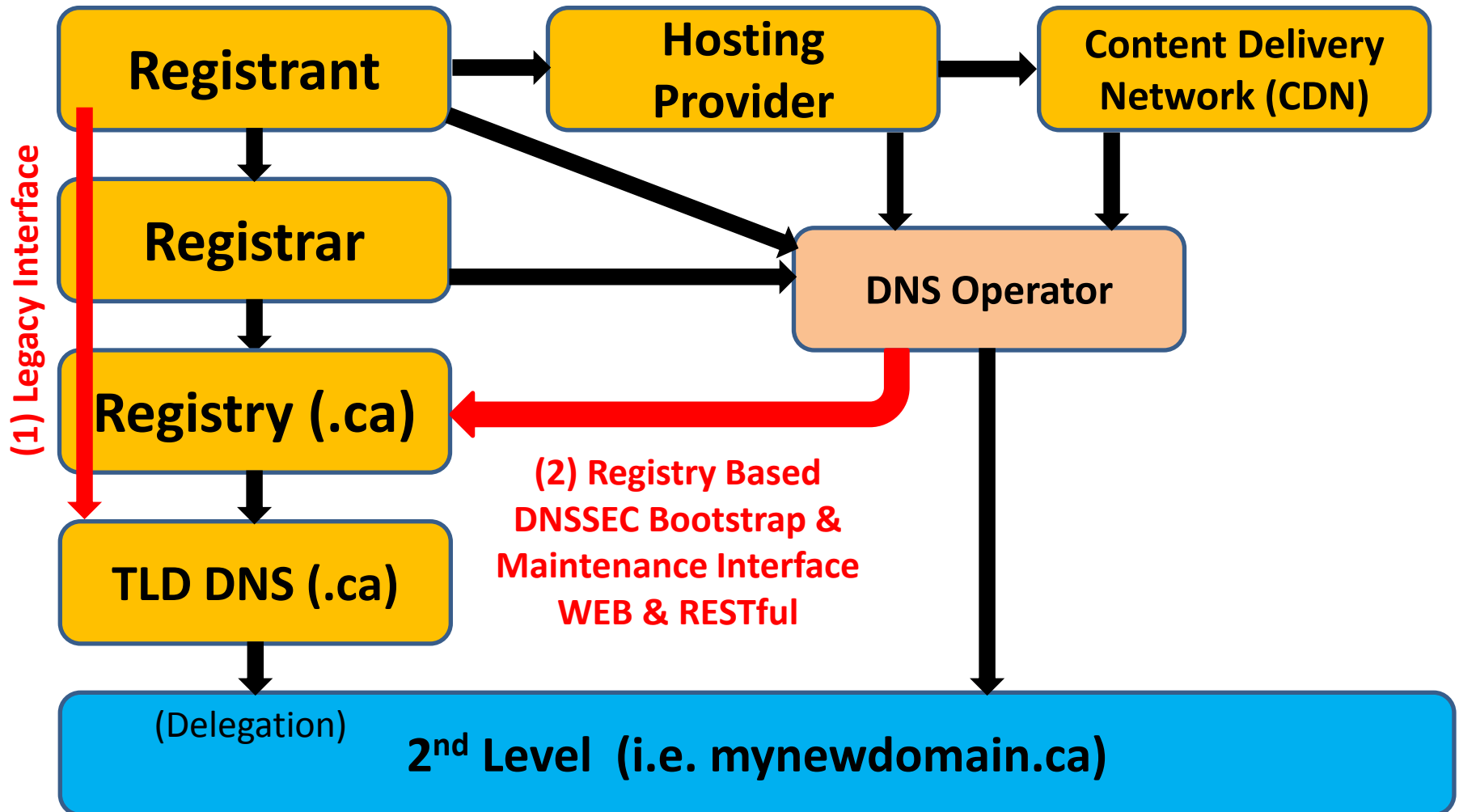
⇒ *All symptoms of a root cause.*

We need to change the model to support different authorization/delegation model for NS/DS/Glue, and a protocol to manage up the food chain.

DNSSEC Provisioning Reality



DNSSEC Provisioning - Proposed



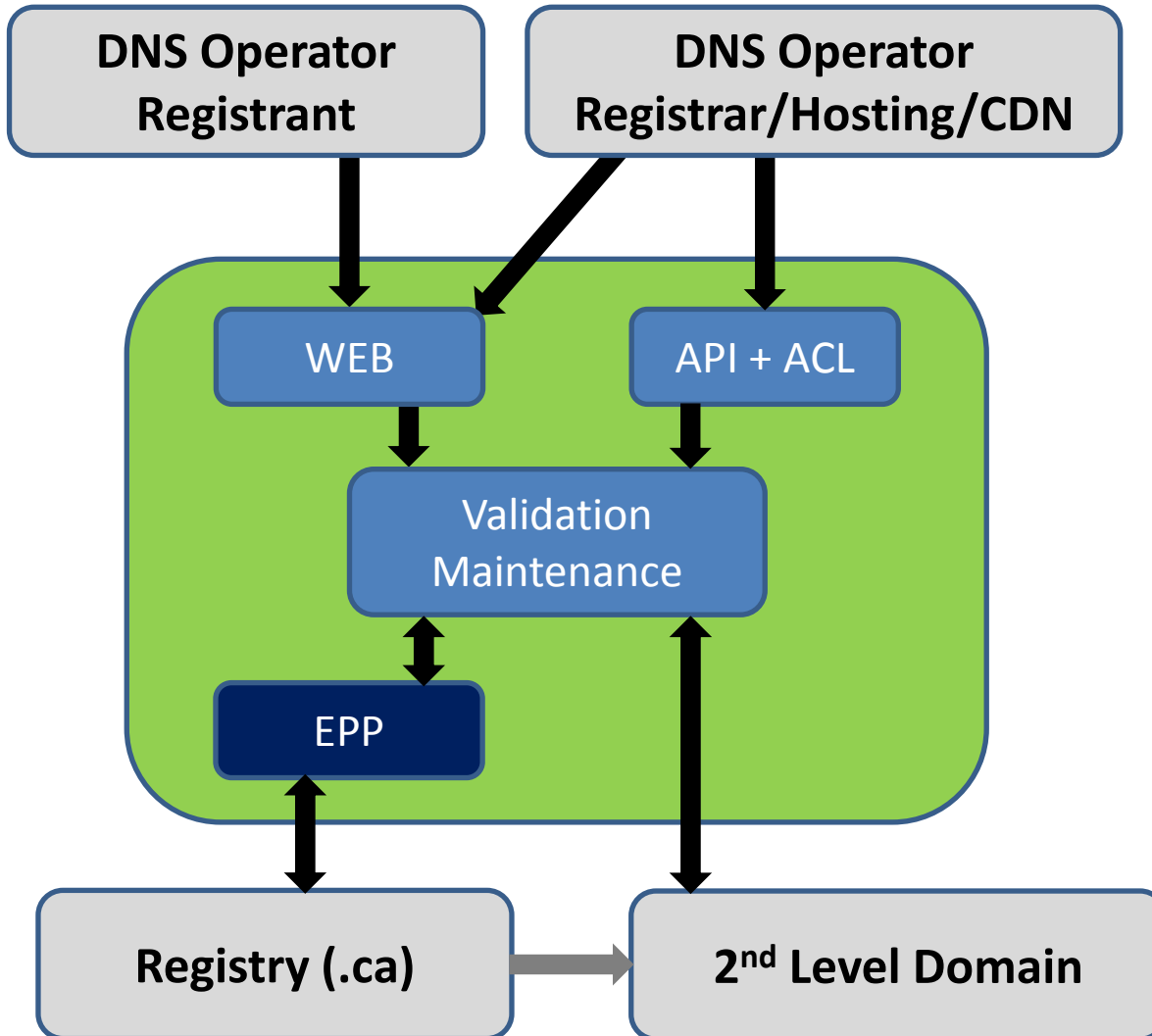
DNSSEC Bootstrap Process

- The DNS Operator needs to prove they control and operate the properly signed and delegated 2nd level zone.
 - Control is proven by adding `_delegate` TXT record(s) with KEYID(s) of DNSKEY to put in the registry
 - Operate is proven by submitting a request at the registry (.ca) via web gui or RESTful API to trigger the bootstrap process.

DNSSEC Validation Process

- The validation process ensures;
 - Over TCP
 - The RRSig signatures are valid
 - The NS RRset at parent and child are valid
 - _delegate TXT records matches DNSKEY
- The process is to make sure it's signed and delegated properly and ready
 - If already bootstrapped then ignore duplicate requests
 - If not signed properly, provide error dump why it failed

DNSSEC Provisioning - Proposed



DNS Operator to prove control of the SLD by publishing a _delegate TXT record with DNSKEY ID.

Validation via TCP will create a DS from DNSKEY if all signatures are good

DS added to .ca zone file via EPP transaction

Maintenance done via polling CDS records

Now what? We Need Prototypes!!!

- The WEB interface is at:
 - <http://cira.nohats.ca>
- The RESTful API interface is at:
 - <http://cira.nohats.ca/gends/>
 - eg: <http://cira.nohats.ca/gends/nohats.ca>
- Yes, needs a bit of security & controls 😊

Web Based Prototype

The image displays two browser windows side-by-side, illustrating a web-based DNSSEC tool. The top window, titled "DNSSEC Start", shows a search interface with a "Domain:" label and an input field containing "nohats.ca", followed by a blue "Go!" button. The bottom window, titled "DNSSEC Result for nohats.ca", shows the results of a DNSSEC lookup. It includes a "Domain:" field with "nohats.ca", a "Result:" field with the DNS record "nohats.ca. 86400 IN DS 27545 8 2 FB64BE1753654F696249D7EA7B80583BFE22BA9E8CD5102BC3E3E11BABD3B75B", and a "Debug:" section containing a detailed log of the lookup process.

DNSSEC Start

Domain:

DNSSEC Result for nohats.ca

Domain:

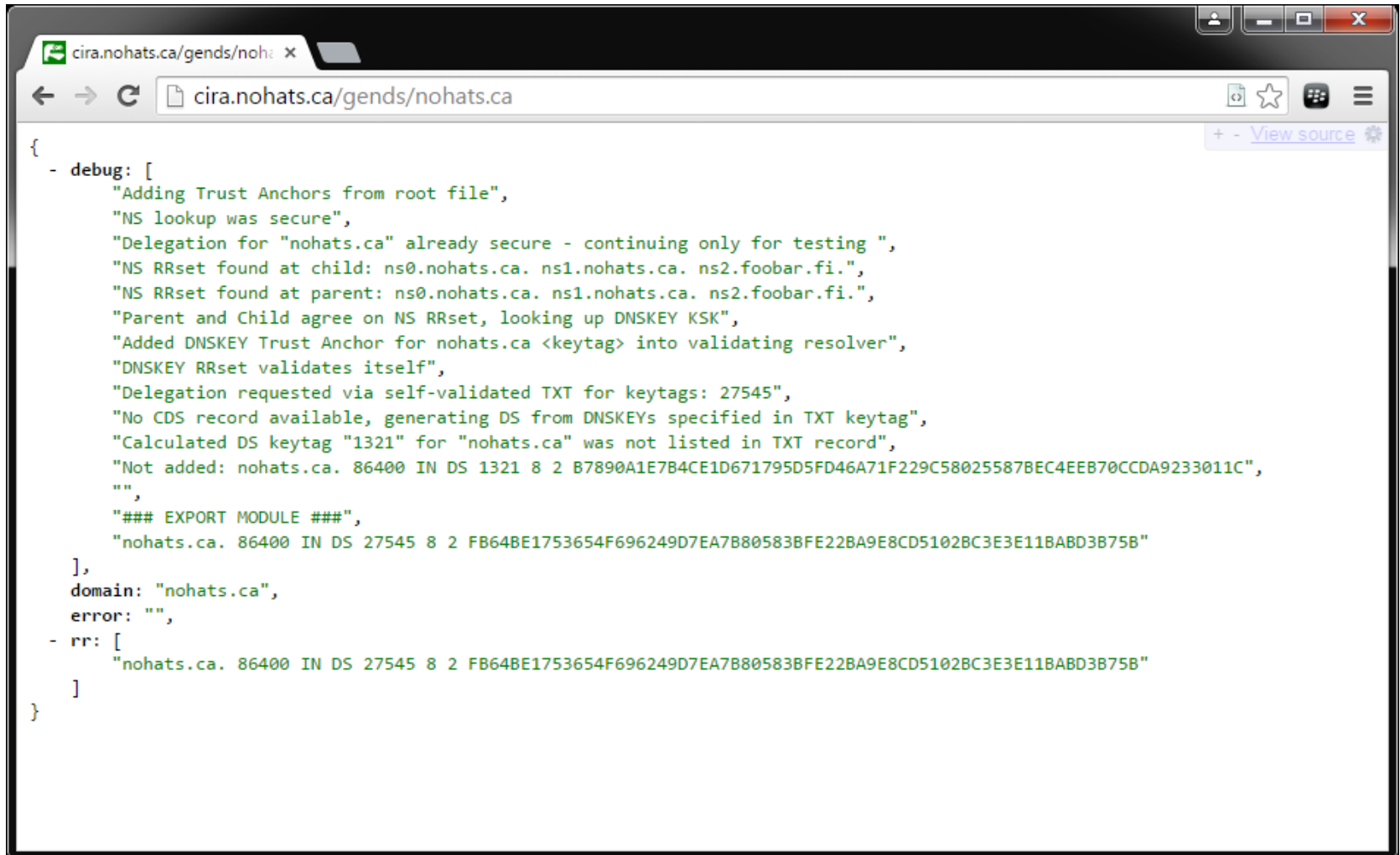
Result:

Debug:

```
Adding Trust Anchors from root file
NS lookup was secure
Delegation for "nohats.ca" already secure - continuing only for testing
NS RRset found at child: ns0.nohats.ca. ns1.nohats.ca. ns2.foobar.fi.
NS RRset found at parent: ns0.nohats.ca. ns1.nohats.ca. ns2.foobar.fi.
Parent and Child agree on NS RRset, looking up DNSKEY KSK
Added DNSKEY Trust Anchor for nohats.ca <keytag> into validating resolver
DNSKEY RRset validates itself
Delegation requested via self-validated TXT for keytags: 27545
No CDS record available, generating DS from DNSKEYs specified in TXT keytag
Calculated DS keytag "1321" for "nohats.ca" was not listed in TXT record
Not added: nohats.ca. 86400 IN DS 1321 8 2 B7890A1E7B4CE1D671795D5FD46A71F229C58025587BEC4EEB70CCDA9233011C

### EXPORT MODULE ###
nohats.ca. 86400 IN DS 27545 8 2 FB64BE1753654F696249D7EA7B80583BFE22BA9E8CD5102BC3E3E11BABD3B75B
```

RESTful API Prototype



The screenshot shows a web browser window with the address bar containing `cira.nohats.ca/gends/nohats.ca`. The page content displays a JSON response from a RESTful API. The response is a JSON object with a `debug` array, a `domain` string, an `error` string, and an `rr` array. The `debug` array contains several log messages detailing the DNS resolution process, including the addition of trust anchors, NS RRset discovery, and the calculation of a DS keytag. The `rr` array contains a single DNS record: `"nohats.ca. 86400 IN DS 27545 8 2 FB64BE1753654F696249D7EA7B80583BFE22BA9E8CD5102BC3E3E11BABD3B75B"`.

```
{
  - debug: [
    "Adding Trust Anchors from root file",
    "NS lookup was secure",
    "Delegation for \"nohats.ca\" already secure - continuing only for testing ",
    "NS RRset found at child: ns0.nohats.ca. ns1.nohats.ca. ns2.foobar.fi.",
    "NS RRset found at parent: ns0.nohats.ca. ns1.nohats.ca. ns2.foobar.fi.",
    "Parent and Child agree on NS RRset, looking up DNSKEY KSK",
    "Added DNSKEY Trust Anchor for nohats.ca <keytag> into validating resolver",
    "DNSKEY RRset validates itself",
    "Delegation requested via self-validated TXT for keytags: 27545",
    "No CDS record available, generating DS from DNSKEYs specified in TXT keytag",
    "Calculated DS keytag \"1321\" for \"nohats.ca\" was not listed in TXT record",
    "Not added: nohats.ca. 86400 IN DS 1321 8 2 B7890A1E7B4CE1D671795D5FD46A71F229C58025587BEC4EEB70CCDA9233011C",
    "",
    "### EXPORT MODULE ###",
    "nohats.ca. 86400 IN DS 27545 8 2 FB64BE1753654F696249D7EA7B80583BFE22BA9E8CD5102BC3E3E11BABD3B75B"
  ],
  domain: "nohats.ca",
  error: "",
  - rr: [
    "nohats.ca. 86400 IN DS 27545 8 2 FB64BE1753654F696249D7EA7B80583BFE22BA9E8CD5102BC3E3E11BABD3B75B"
  ]
}
```

Maintenance Approach - CDS Record?

- The .ca Registry will take care of performing on-going DNSSEC maintenance of signed domains.
 - Daily (or specific frequency) polling for new CDS RR
 - Manage as per .ca DNSSEC policy (# keys, DS, Algo, etc...)
 - TBD: 48 hours hold + notify admin/tech contacts?
 - .ca controls the DS format... Create new DS when value in CDS are not compliant
- Testing CDS records for on-going maintenance

```
[root@fedora ~]# dig cds demo.nohats.ca +short  
58691 8 2 B5B99B5FBAA7565C49710DCF21137E69EF996C1FC04903BAB4B9397E 5D1BCB09
```

Strategy

- Continue framework development
 - How to maintain and un-sign a domain?
 - Gather & include feedback
- Looking to implement with .ca partners, DNS appliance, Registrars and CDN providers.
- Make code Open Source for all to use
- Standardize - develop new IETF RFC
- Separate DNSSEC from standard registration
 - Investigate registry lock integration/options/value
- Make the Internet a better place 😊

Thank you!

DNSSEC Coordination
<dnssec-coord@elists.isoc.org>