draft-thomassen-generalised-dns-notify-01

Synchronisation and efficency in a distributed ecosystem

Peter Thomassen Johan Stenstam

March 29, 2023

Problem Statement

The (relatively new) RR types `CDS` and `CSYNC` rely on "someone" (typically the parent registry or the registrar) periodically **scanning** some subset of the child zones in search of indications that the child wants to update information in the parent zone.

- Scanning is resource consuming and costly.
- Scanning is inefficient (casting millions of nets return a very small number of fish).
- Slow scanning delays convergence.

There is also a related problem in the context of so-called "multi-signer" setups.

- In that case the scanning is needed to catch changes to the `DNSKEY` RRset, because keys needs to be kept in sync across multiple "signers".

Is There Any "Prior Art"?

Yes, there is.

We've seen an eerily similar movie before: secondaries polling the primaries for the SOA serial to know whether to request a zone transfer or not.

- **That** movie had a happy ending: RFC 1996 (NOTIFY).
- RFC 1996 defines the NOTIFY message, which is sent from a "**primary**" to a "**secondary**" to inform the latter that the contents of a zone has been updated.

Would it be possible to re-use RFC 1996 also for these new use cases?

RFC 1996 To The Rescue

Today an RFC 1996 NOTIFY message always contains an `SOA` record.

- We refer to this as NOTIFY(`SOA`).
- We realised that **RFC 1996 doesn't say that it has to be an** `SOA`. It is just that no one has found any use for anything else.

Our proposal is therefore a generalisation of the original NOTIFY(`SOA`) from RFC 1996 to also define:

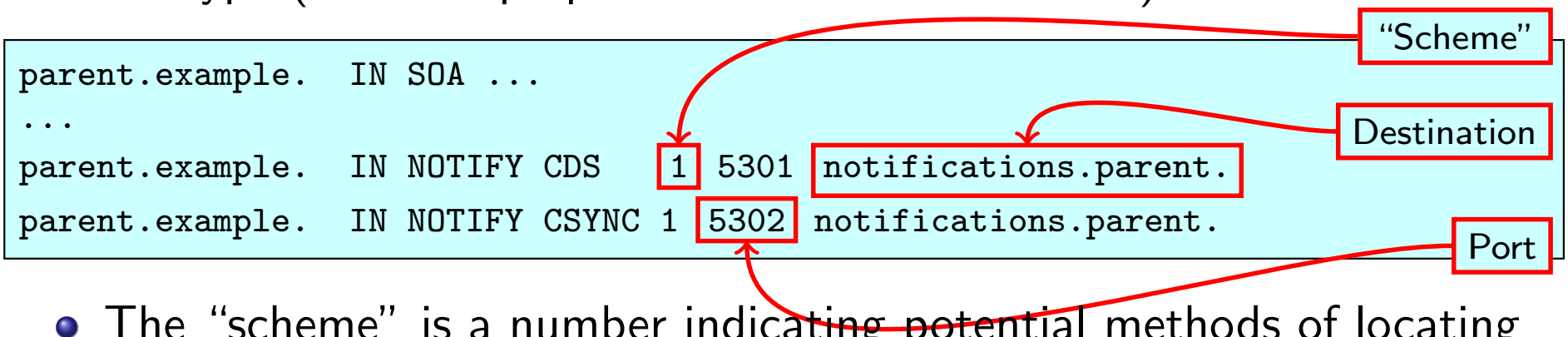
- NOTIFY(`CDS`): Inform the parent that the child has published a new `CDS` RRset.
- NOTIFY(`CSYNC`): Inform the parent that the child has published a new `CSYNC` record.

The intent is to optimise the slow and resource consuming `CDS` and `CSYNC` scanning that an increasing number of TLD registries do.

Where should these NOTIFY(RRtype) Be Sent?

Good question. The NOTIFY(`CDS`) and NOTIFY(`CSYNC`) are "vertical" (from the child to the parent). So a logical choice is to look for locator information in the parent zone.

After a lot of pondering we believe that the best alternative is to specify a new RR type (with the proposed mnemonic "`NOTIFY`"):



- The "scheme" is a number indicating potential methods of locating where to send a NOTIFY. Only the value "`scheme=1`" is defined in the draft, with the method "send the NOTIFY to the specified port at the specified destination".
- Other schemes may be defined in the future.

Short Background: What is "Multi-Signer"?

The idea with multi-signer is that for a DNSSEC-signed zone the signing operation constitute a **SPOF** (single point of failure). It doesn't matter how many anycast nameservers there are if the signing of the zone has failed.

- See the `.SE` well-publicised signing incident from February 2022...

While signing a zone is a complex process it is by now usually 100% automated.

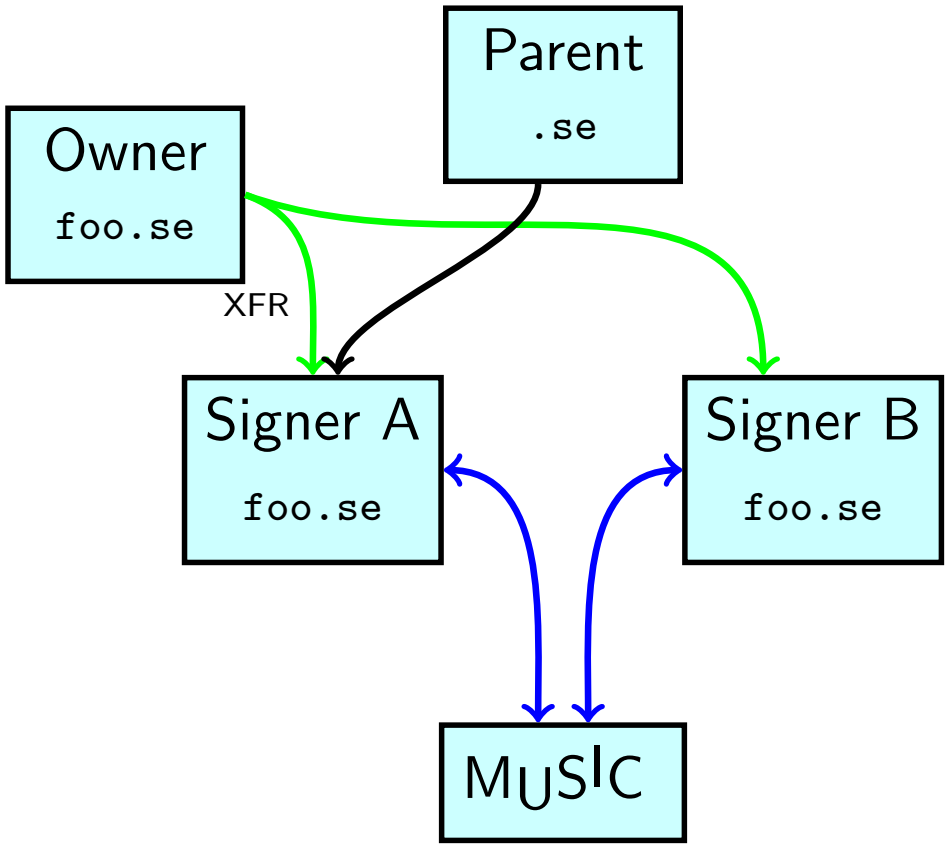
- But as long as this, admittedly nice and shiny, automated signing only occur in one place the SPOF remains.

"Multi-signer" is an attempt at addressing this SPOF, at the cost of new complexity, which will be managed by more automation.

The "Multi-Signer Model", cont'd.

In one implementation there is a "MUlti-SIgner Controller", aka MUSIC.

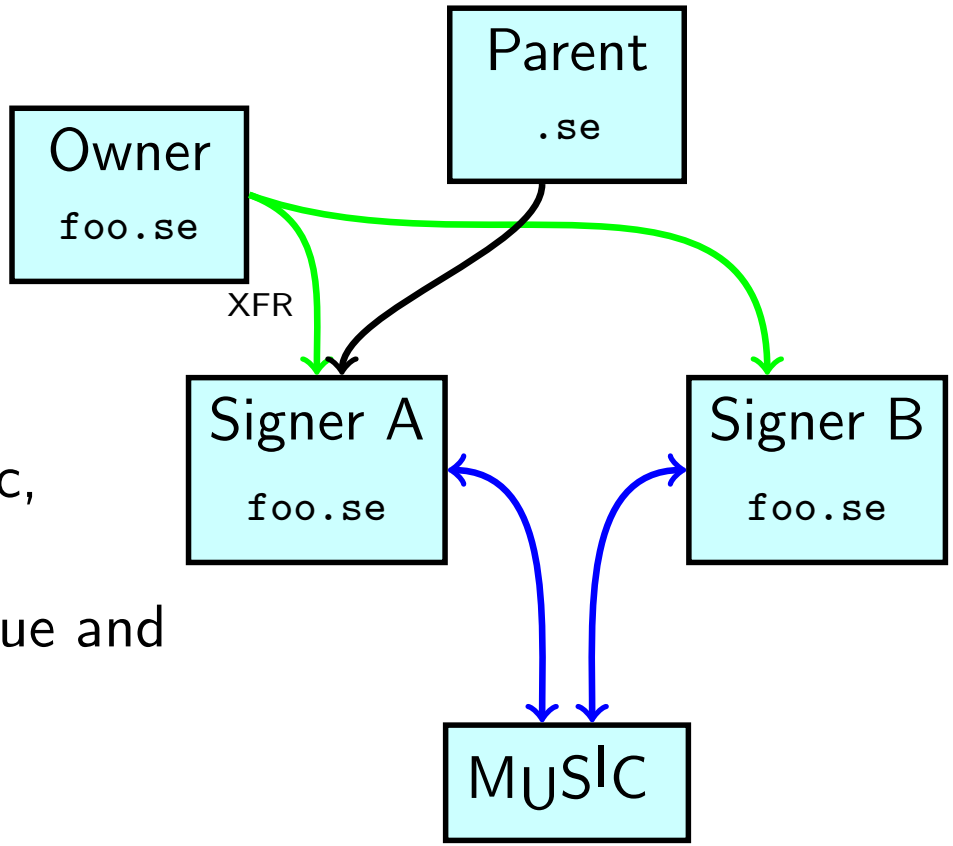
- The zone owner sends the unsigned zone to all **signers**.
- All **signers** sign the zone with own keys.
- MUSIC talks to all signers and gets data that affects DNSSEC (`DNSKEY`s and then `CDS`) in sync.



The "Multi-Signer Model", cont'd.

In one implementation there is a "MUlti-SIgner Controller", aka MUSIC.

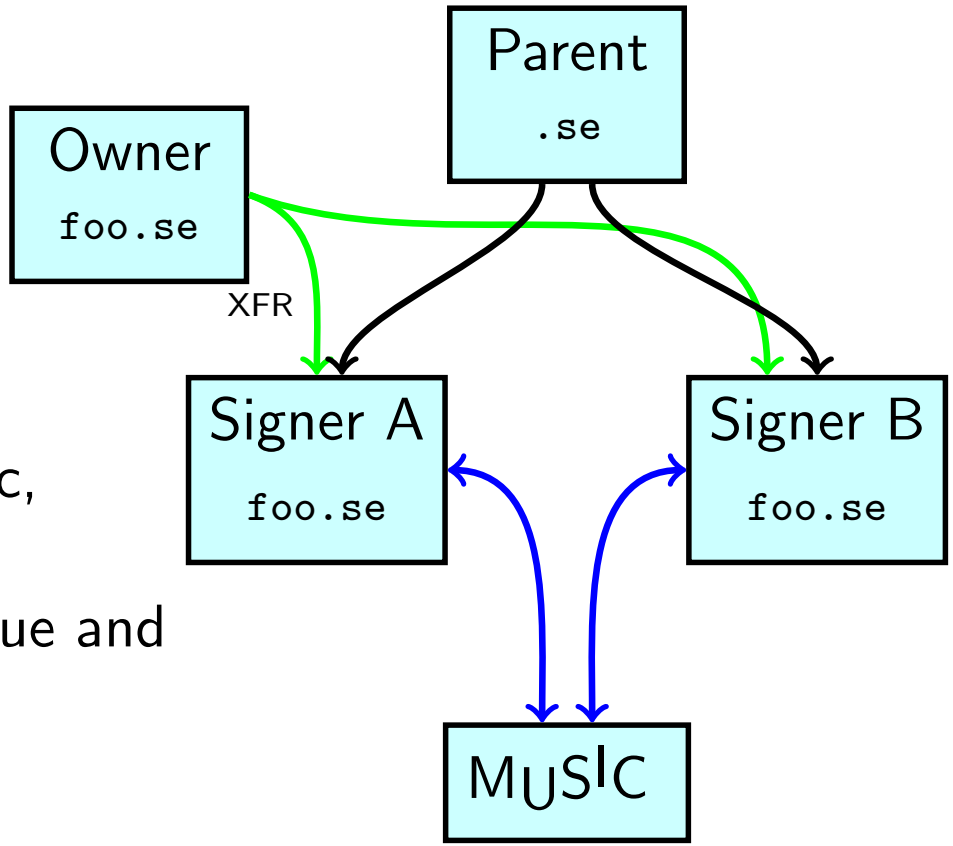
- The zone owner sends the unsigned zone to all **signers**.
- All **signers** sign the zone with own keys.
- MUSIC talks to all signers and gets data that affects DNSSEC (`DNSKEY`s and then `CDS`) in sync.
- When DNSSEC data is in sync, MUSIC talks to signers to get delegation info in sync (`NS`, glue and then `CSYNC`).



The "Multi-Signer Model", cont'd.

In one implementation there is a "MUlti-SIgner Controller", aka MUSIC.

- The zone owner sends the unsigned zone to all **signers**.
- All **signers** sign the zone with own keys.
- MUSIC talks to all signers and gets data that affects DNSSEC (`DNSKEY`s and then `CDS`) in sync.
- When DNSSEC data is in sync, MUSIC talks to signers to get delegation info in sync (`NS`, glue and then `CSYNC`).



What Are The Challenges With This Model?

- . . . modulo bugs: If the model is viable we will get robust implementations over time.
- The primary issue with the current multi-signer model is that DNSSEC key rollovers are mostly automatic. ZSK rollovers fully so, and KSK rollovers getting there.
  - And the signers consider ZSK rollovers to be a completely internal operation. . . so they will not inform anyone.
  - When **Signer A** rolls the ZSK the `DNSKEY` RRset is affected. . .
  - . . . which must be kept in sync with **Signer B**.
  - . . . because if **Signer B**'s KSK has not signed the new **Signer A** ZSK it isn't good.

What Are The Challenges With This Model?

- . . . modulo bugs: If the model is viable we will get robust implementations over time.
- The primary issue with the current multi-signer model is that DNSSEC key rollovers are mostly automatic. ZSK rollovers fully so, and KSK rollovers getting there.
  - And the signers consider ZSK rollovers to be a completely internal operation. . . so they will not inform anyone.
  - When **Signer A** rolls the ZSK the `DNSKEY` RRset is affected. . .
  - . . . which must be kept in sync with **Signer B**.
  - . . . because if **Signer B**'s KSK has not signed the new **Signer A** ZSK it isn't good.

The problem is simply that as **Signer A** today doesn't inform anyone about the key rollover a window of vulnerability opens until MUSIC (or some other controller) notices.

RFC 1996 To The Rescue. Again.

We propose that in addition to the

- `NOTIFY(SOA)` from RFC 1996
- `NOTIFY(CDS)` and `NOTIFY(CSYNC)` above

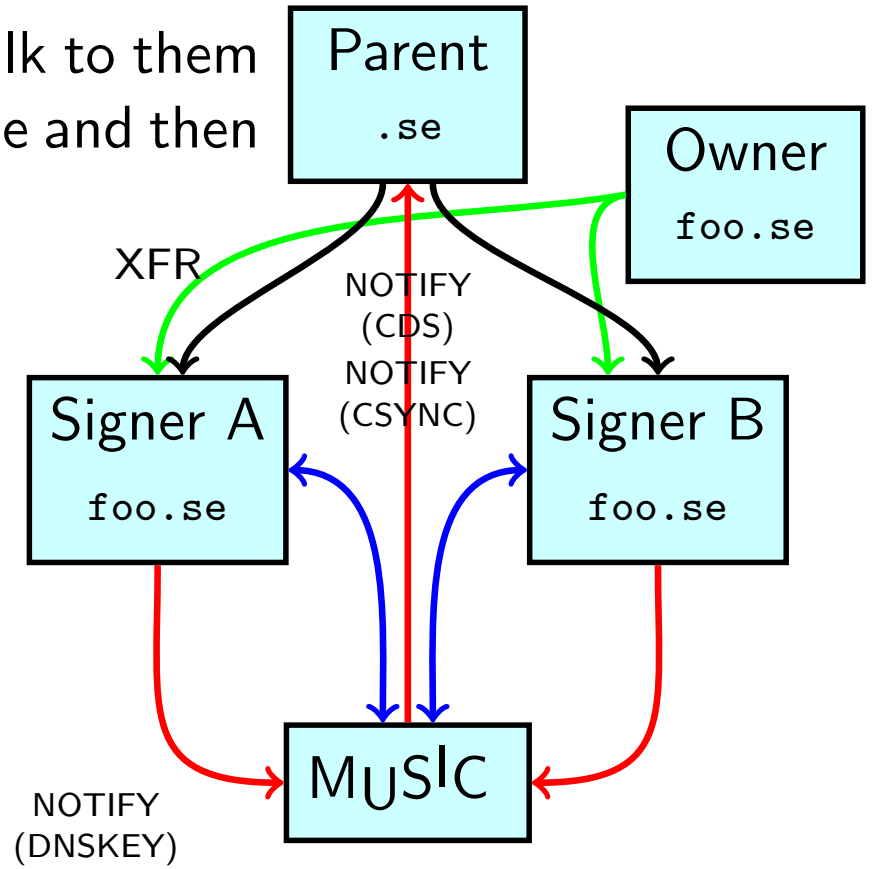
yet another NOTIFY is defined:

- `NOTIFY(DNSKEY)`: Inform the owner, or its designated controller, that a new `DNSKEY` RRset has been (or is about to be) published.

This is a mechanism to enable **signers** to, in time, inform about a planned key rollover and thereby collapse the window of vulnerability caused by having the `DNSKEY` RRsets temporarily out-of-sync among multiple signers.

The "Multi-Signer Model", With Notifications

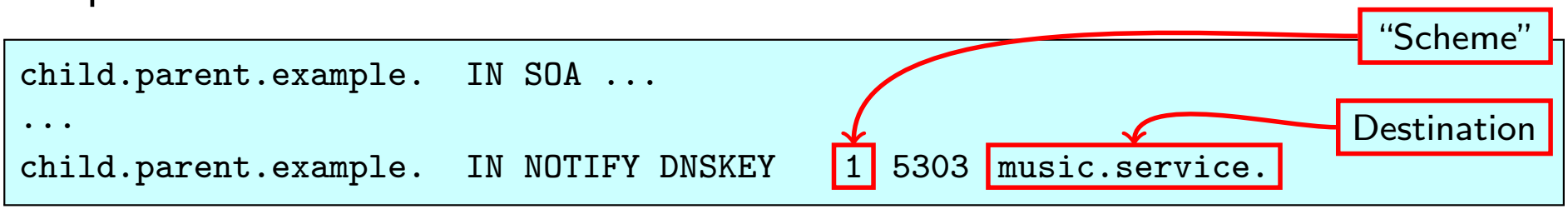
- The zone owner sends the unsigned zone to all **signers**.
- MUSIC talks to signers and gets data that affects DNSSEC in sync (DNSKEYs and then CDS).
- When DNSSEC is in sync, MUSIC talk to them to get delegation info in sync (NS, glue and then CSYNC).
- When MUSIC adds a CDS or CSYNC record it will also send the corresponding NOTIFY(CDS) or NOTIFY(CSYNC) to the **parent**.
- When a signer rolls keys it will send a NOTIFY(DNSKEY) to MUSIC.



Where should NOTIFY(DNSKEY) Be Sent?

Another good question. In this case it isn't to the parent ("vertically"), it is more "sideways", eg. to a multi-signer controller.

So the location information can be located in the child zone, instead of in the parent zone:



- Only the value "scheme=1" is defined in the draft, with the method "send the NOTIFY(DNSKEY) to the specified port at the specified destination".
- Other schemes may be defined in the future.

How Would This Work in an RRR Model?

Under an RRR model it may not be possible for the registry to make updates to the registrant data. Sometimes updates may have to be done by the registrar, typically via EPP.

- If not to the parent, then it is not obvious where to send the NOTIFY(`CDS`) and/or NOTIFY(`CSYNC`) given lots of registrars and no single place to look (like `parent.example. IN NOTIFY` in our proposal).

This is obviously an important issue for generalised DNS notifications to be viable for the entire name space and therefore requires more community input. We have one observation and two suggestions:

- **Observation:** The parent knows who the registrar is for each zone. Nothing precludes the registry (parent) and registrar to sort out where (and if) each registrar would like to receive notifications.
- **Suggestion:** Forward the NOTIFY (via DNS, via EPP, whatever).
- **Suggestion:** The `scheme` parameter may be used in the future to define other mechanisms of locating where to send notifications.

What About The Security Model?

The security model is suggested to be the same as for RFC 1996 NOTIFY.

- I.e. a generalised NOTIFY does not change the verification logic in the recipient (like a scanner), it is only a hint that this particular child zone likely has recently published a `CDS` (or `CSYNC`).
- The point of the hint is to speed up convergence and thereby provide a better service to the child zone, not to change the verification logic.

It is also worth pointing out that a difference from RFC 1996 is that while NOTIFY(`SOA`) are sent **to a nameserver**,

- NOTIFY(`CDS`), etc, are sent **to a service**, be it to a scanner service or to a multi-signer controller (or something else).
- While the packet format is "DNS", it is not communication between nameservers and for this reason we propose not using port 53.

Summary

- By generalising RFC 1996 NOTIFY(SOA) with NOTIFY(CDS) and NOTIFY(CSYNC) we believe that efficiency of CDS and CSYNC scanning may be improved significantly.
  - Both in the resource requirements, but primarily via faster convergence.
- By the further generalisation of NOTIFY(DNSKEY) a signifcant (almost show-stopper) issue in the multi-signer effort gets a clean solution and path forward.
- Generalising RFC1996 (NOTIFY) **does not constitute a protocol change**. This is already allowed (and works fine).
- The proposed location mechanism (eg. defining a new NOTIFY record type used to locate the recipient of a generalised NOTIFY) would be a protocol change.

We would like this work to be adopted by the working group.

Contact Information

Peter Thomassen	peter@desec.io
Johan Stenstam	johan.stenstam@internetstiftelsen.se
Working code	https://github.com/johanix/gen-notify-test