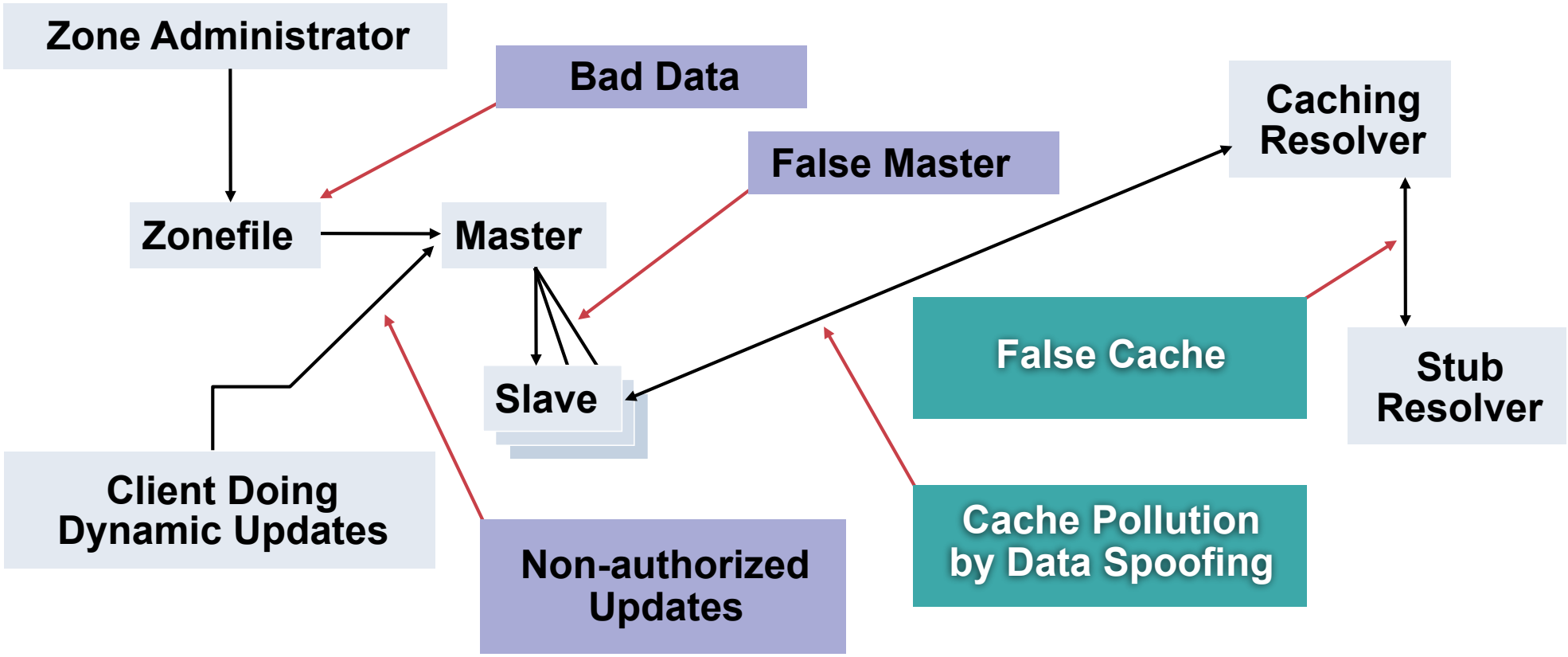
THE BRUTAL WORLD OF DNSSEC

Patrik Fältström

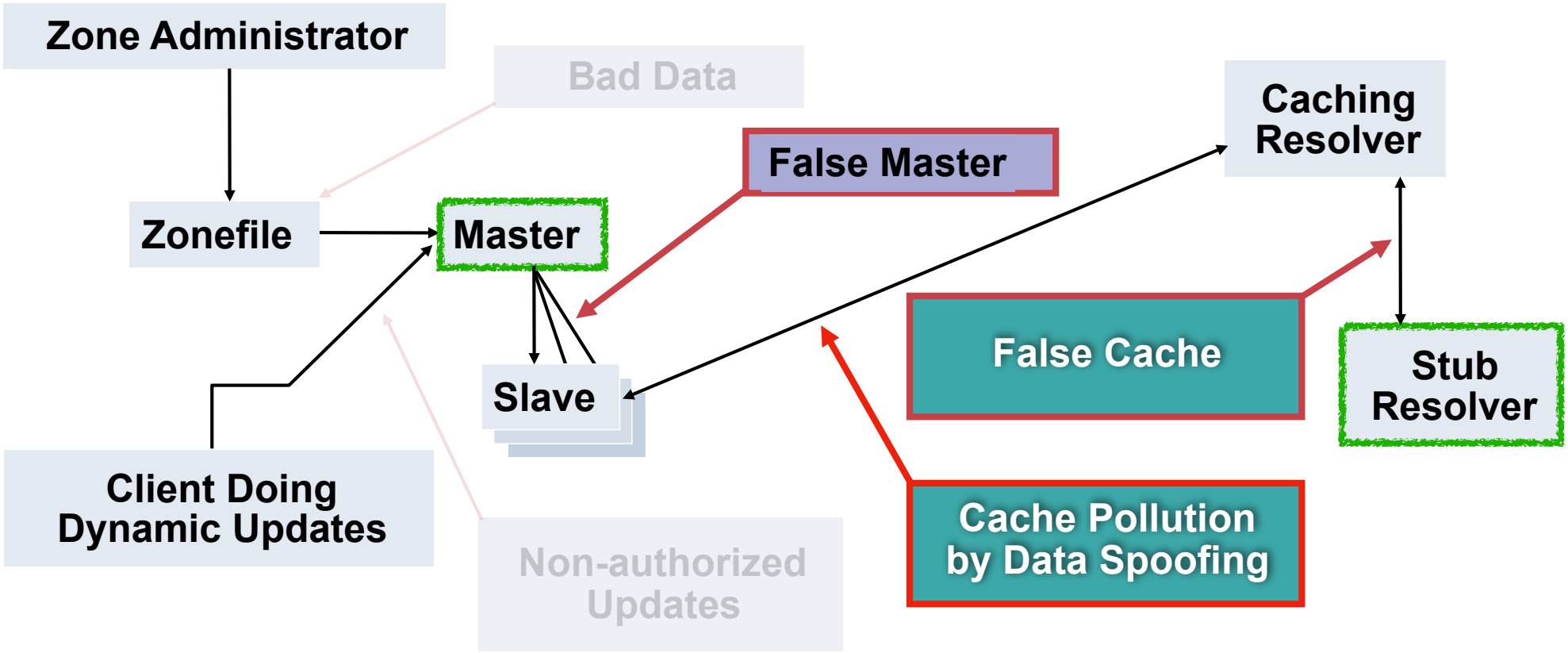
Head of Technology

Netnod

Security Issues with DNS



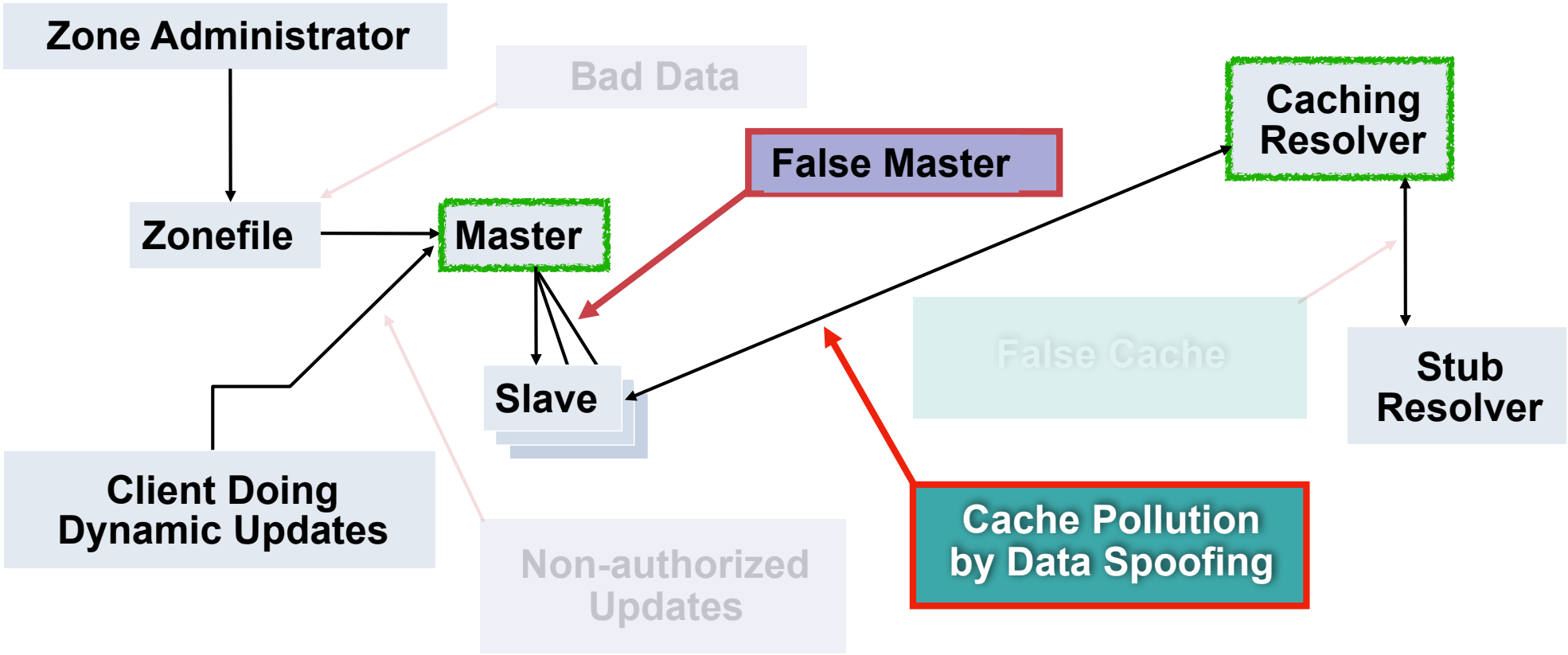
Security Issues with DNS



Protection of Server Data

Protection of Data

Security Issues with DNS



Protection of Server Data

Protection of Data

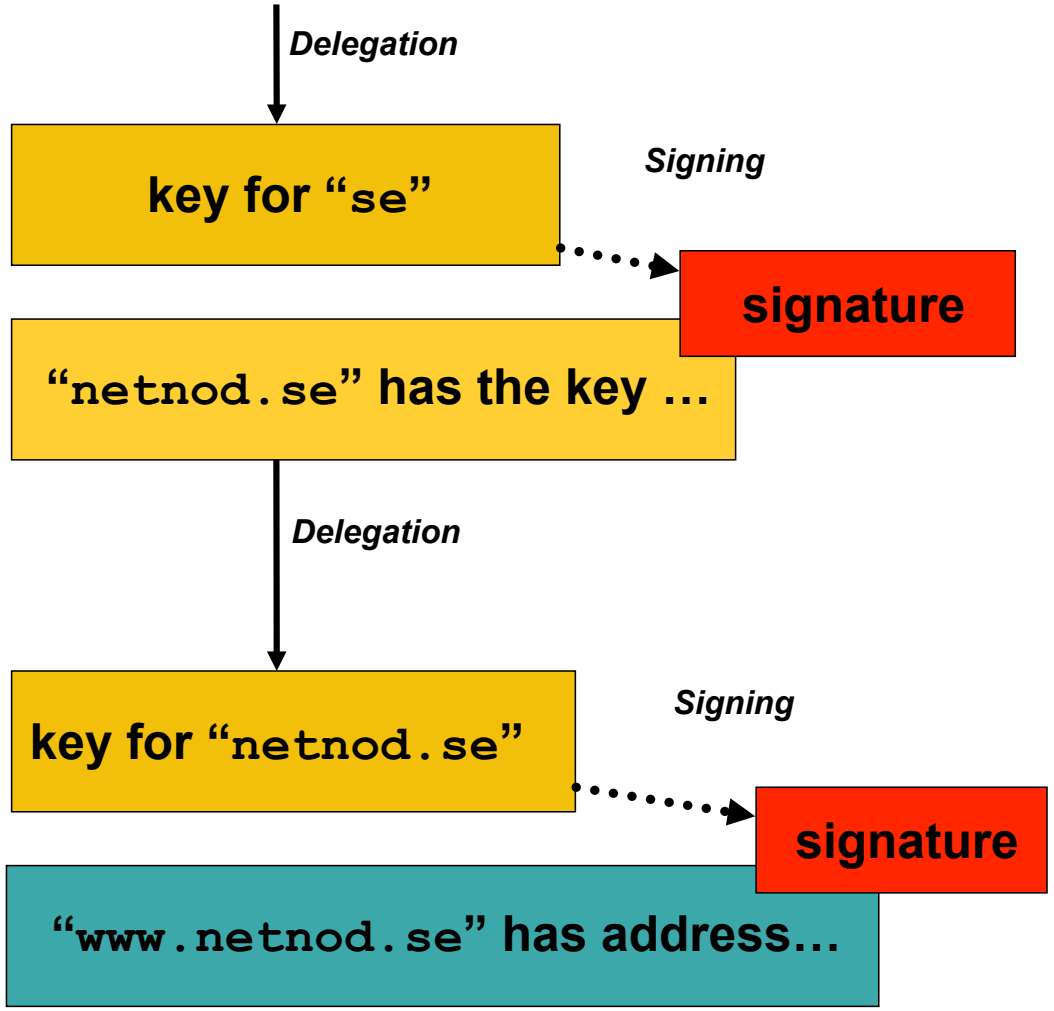
Protection of DNS data

Sign keys stored in DNS

One parent signs child's key

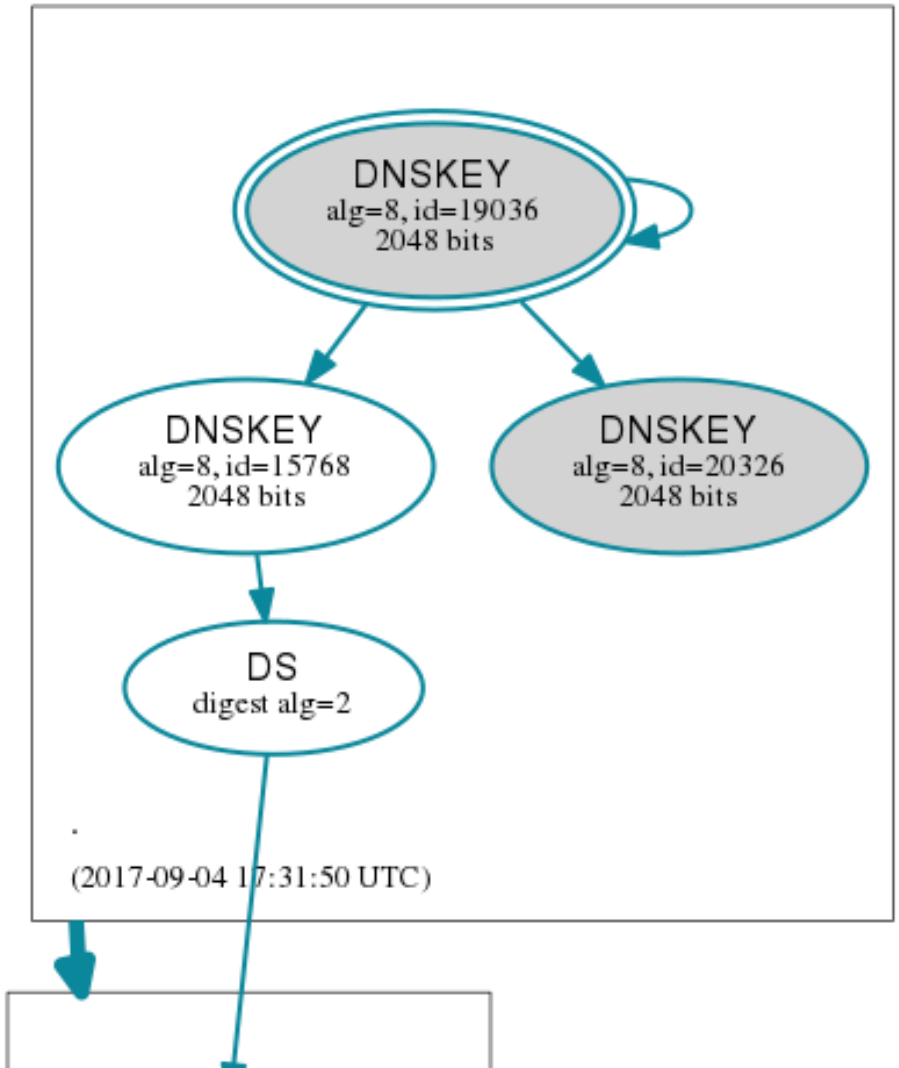
It is sufficient to have the key for "se" or even "."

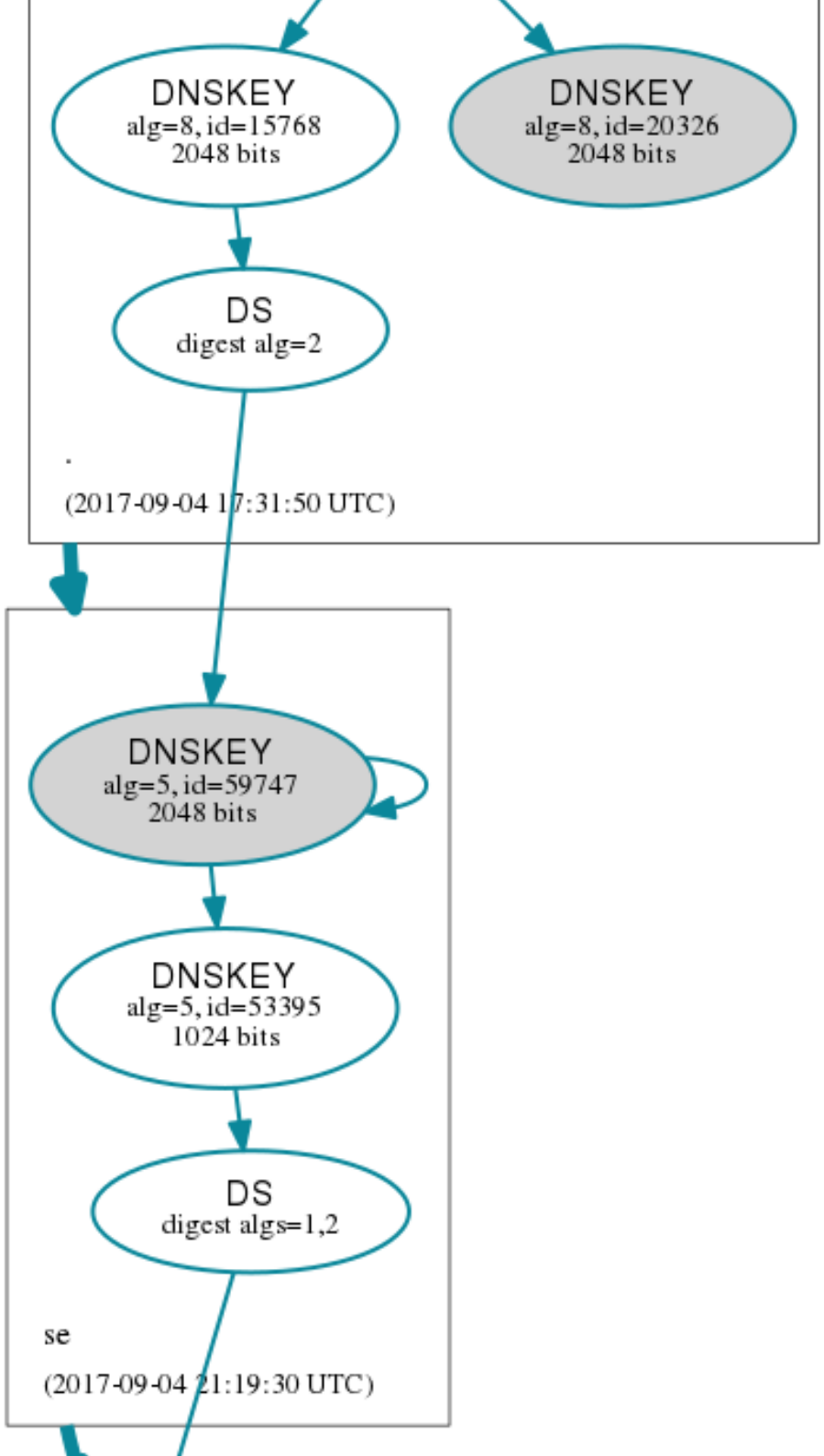
To avoid being spoofed the "top most" key must be acquired by some "outside" method (i.e. not just look it up in DNS)

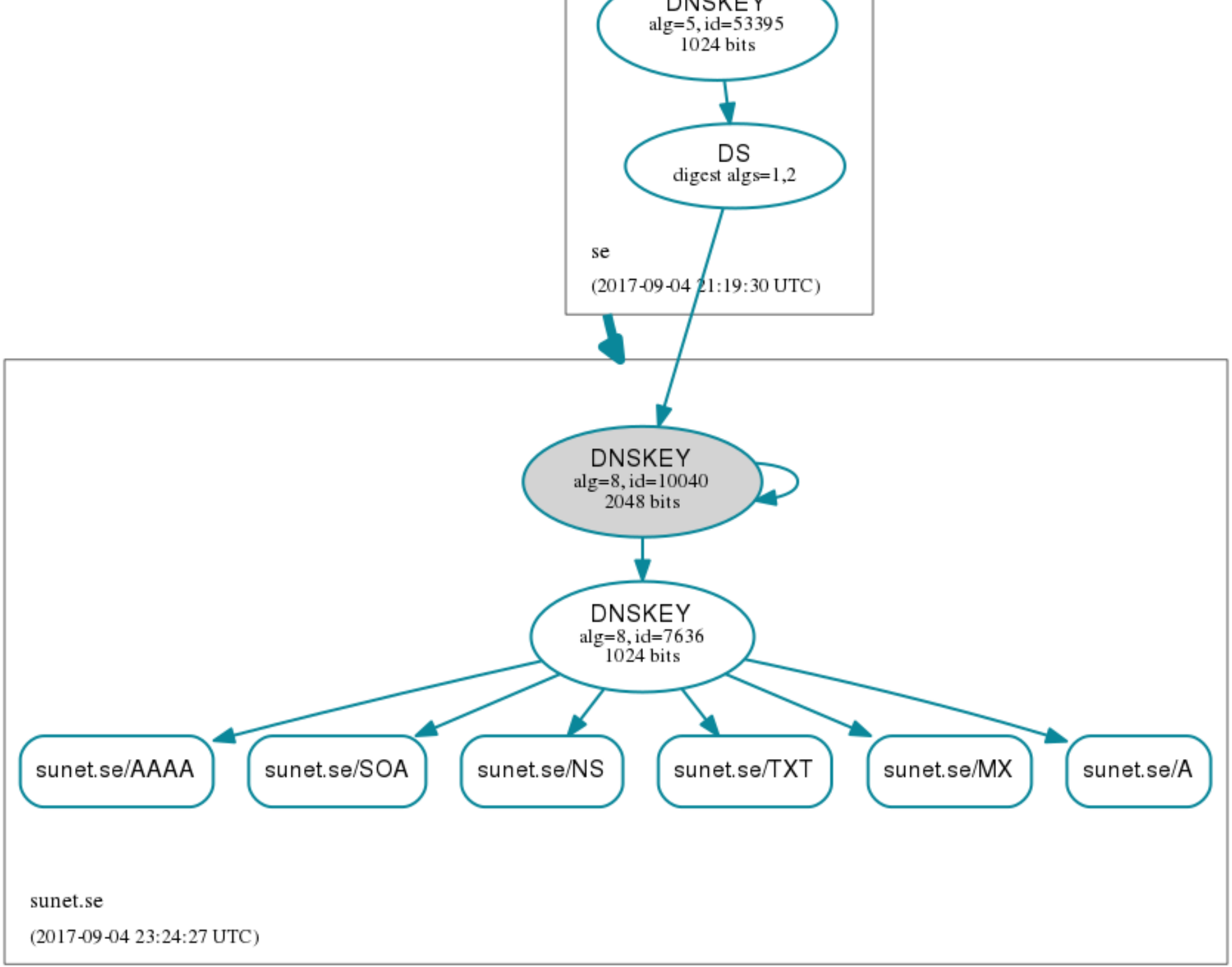


Chain of trust from root to sunet.se

http://dnsviz.net/



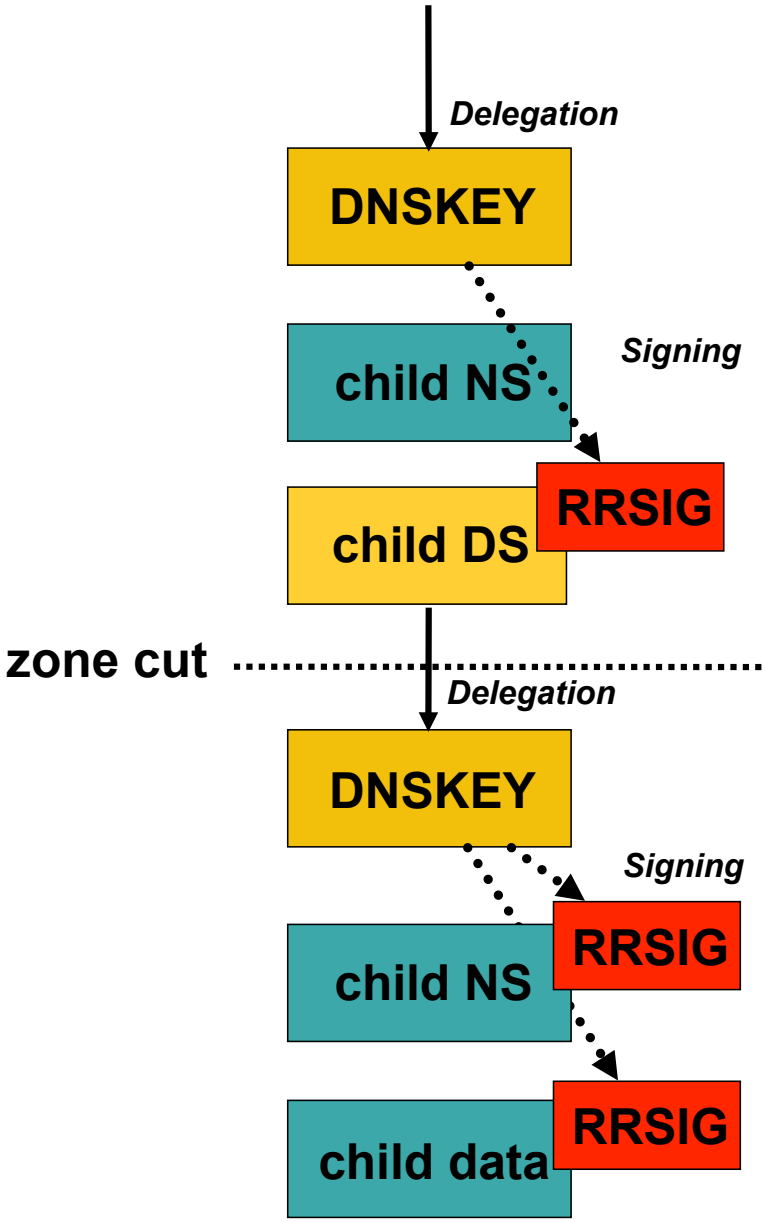

DNSKEY
alg=8, id=15768
2048 bits
DNSKEY
alg=8, id=20326
2048 bits
DS
digest alg=2
.
(2017-09-04 17:31:50 UTC)
DNSKEY
alg=5, id=59747
2048 bits
DNSKEY
alg=5, id=53395
1024 bits
DS
digest algs=1,2
se
(2017-09-04 21:19:30 UTC)


DNSKEY
alg=5, id=53395
1024 bits
DS
digest algs=1,2
se
(2017-09-04 21:19:30 UTC)
DNSKEY
alg=8, id=10040
2048 bits
DNSKEY
alg=8, id=7636
1024 bits
sunet.se/AAAA
sunet.se/SOA
sunet.se/NS
sunet.se/TXT
sunet.se/MX
sunet.se/A
sunet.se
(2017-09-04 23:24:27 UTC)

DS - Delegation Signer

With Delegation Signer a delegation is changed to contain

- ✓ NS records for the child and possible glue (as usual)
- ✓ a secure hash of the child's "key signing key" (this is the DS record)
- ✓ an RRSIG generated by a parent key of the hash (i.e. of the DS record) to prove that the DNS record is authenticated by the parent



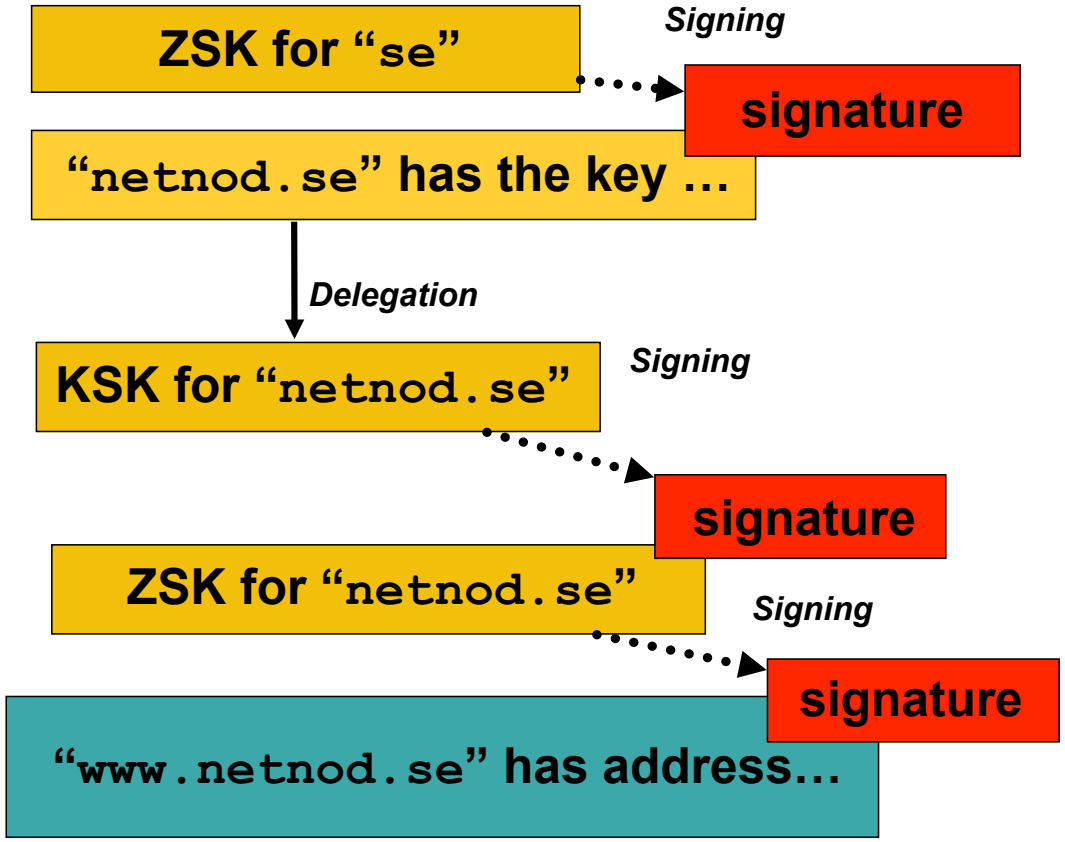
Zone cut with KSK and ZSK

Parent ZSK sign DS record in parent zone

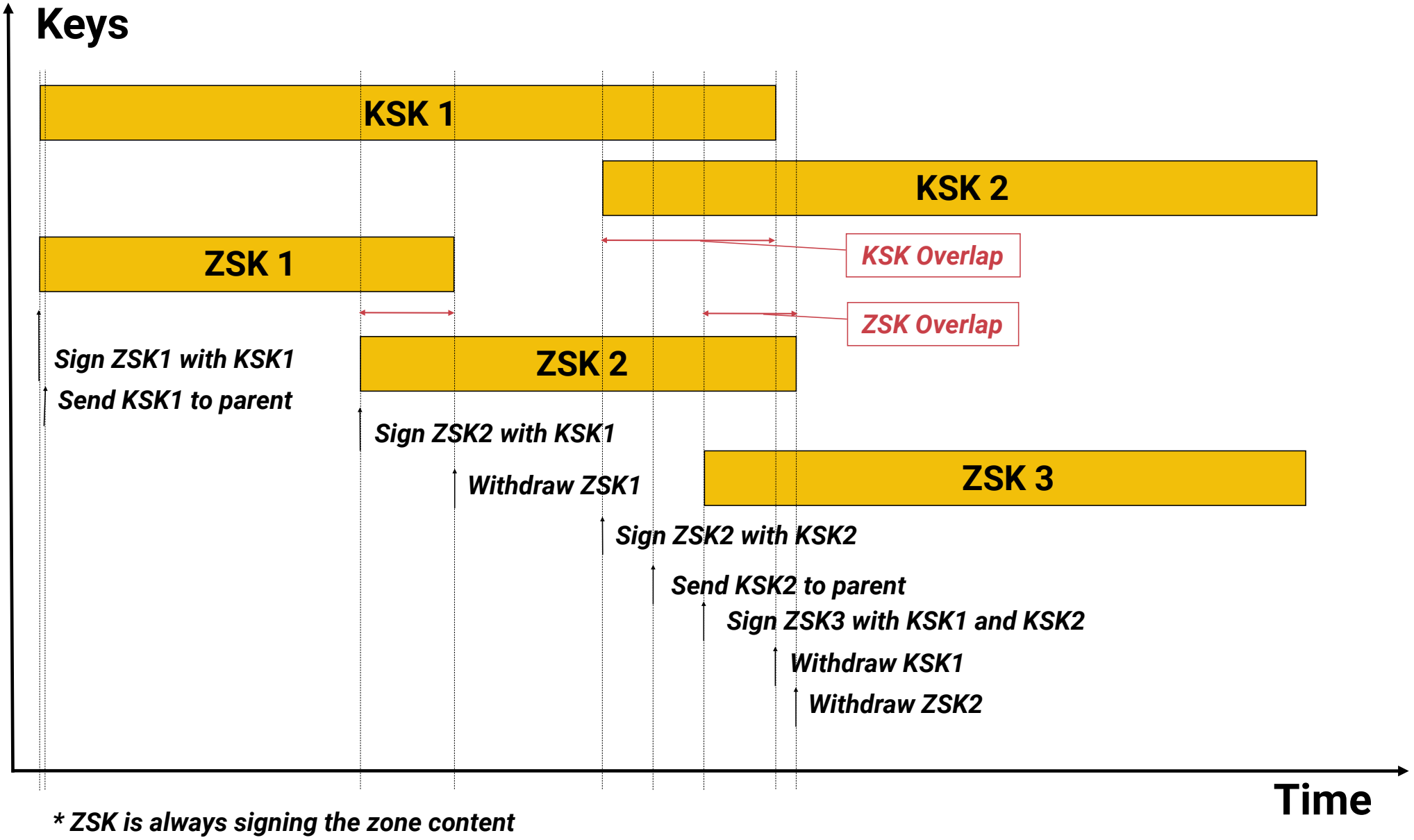
DS record identify KSK for child zone

KSK sign ZSK record in child zone

ZSK sign zone content

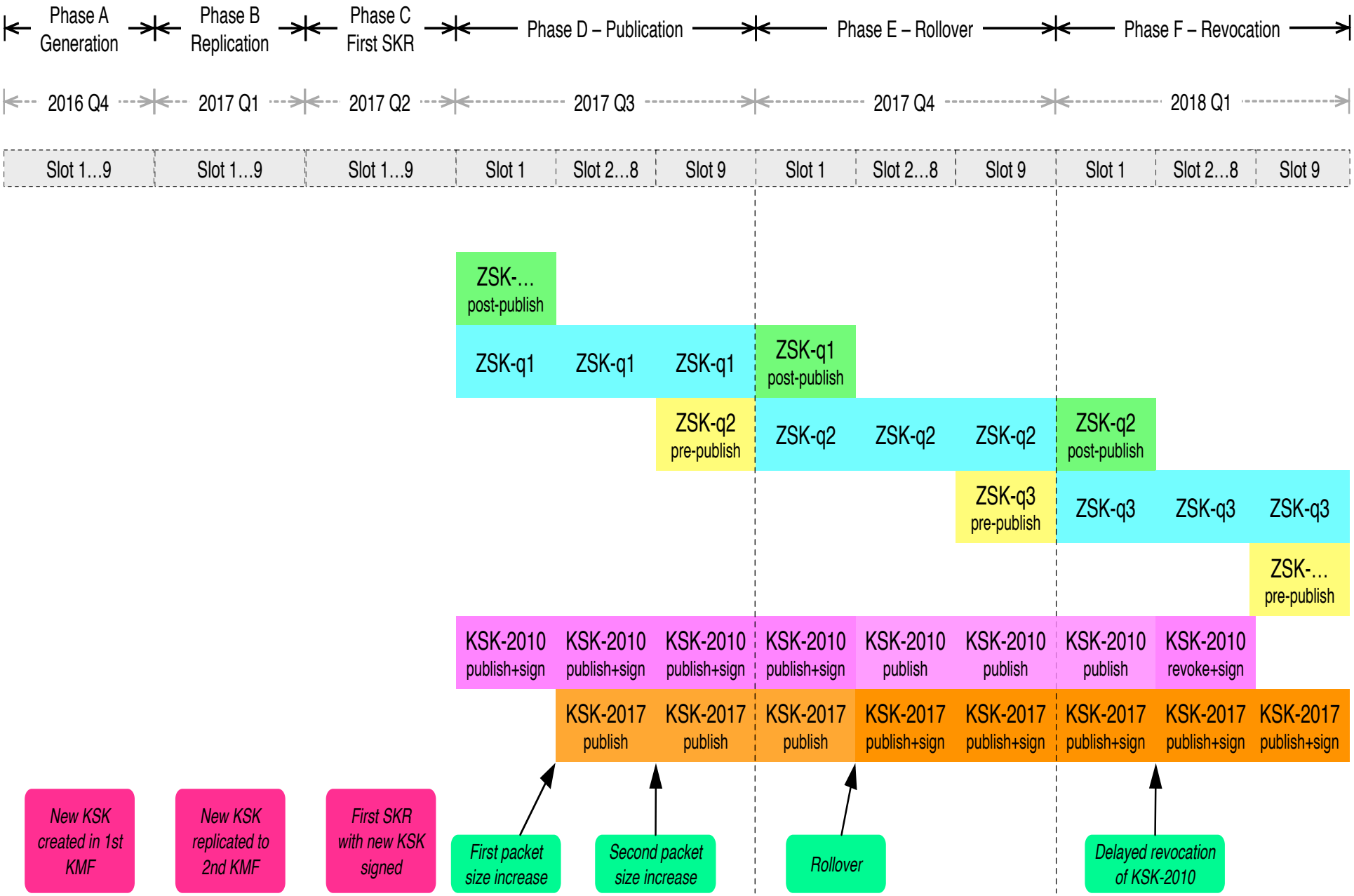


Key rollover timeline



** ZSK is always signing the zone content*

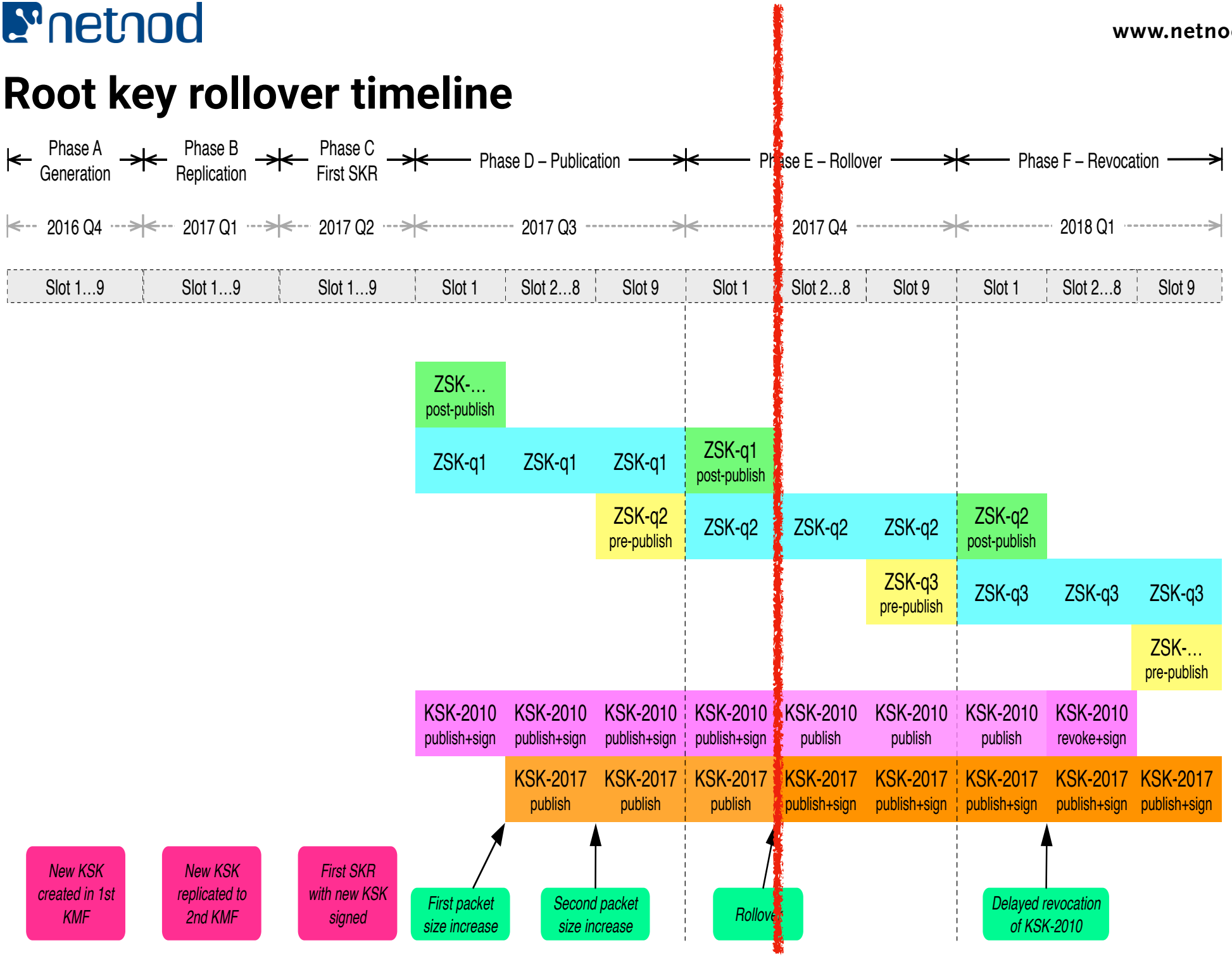
Root key rollover timeline



netnod

www.netnod.se

Root key rollover timeline



Root zone KSK rollover

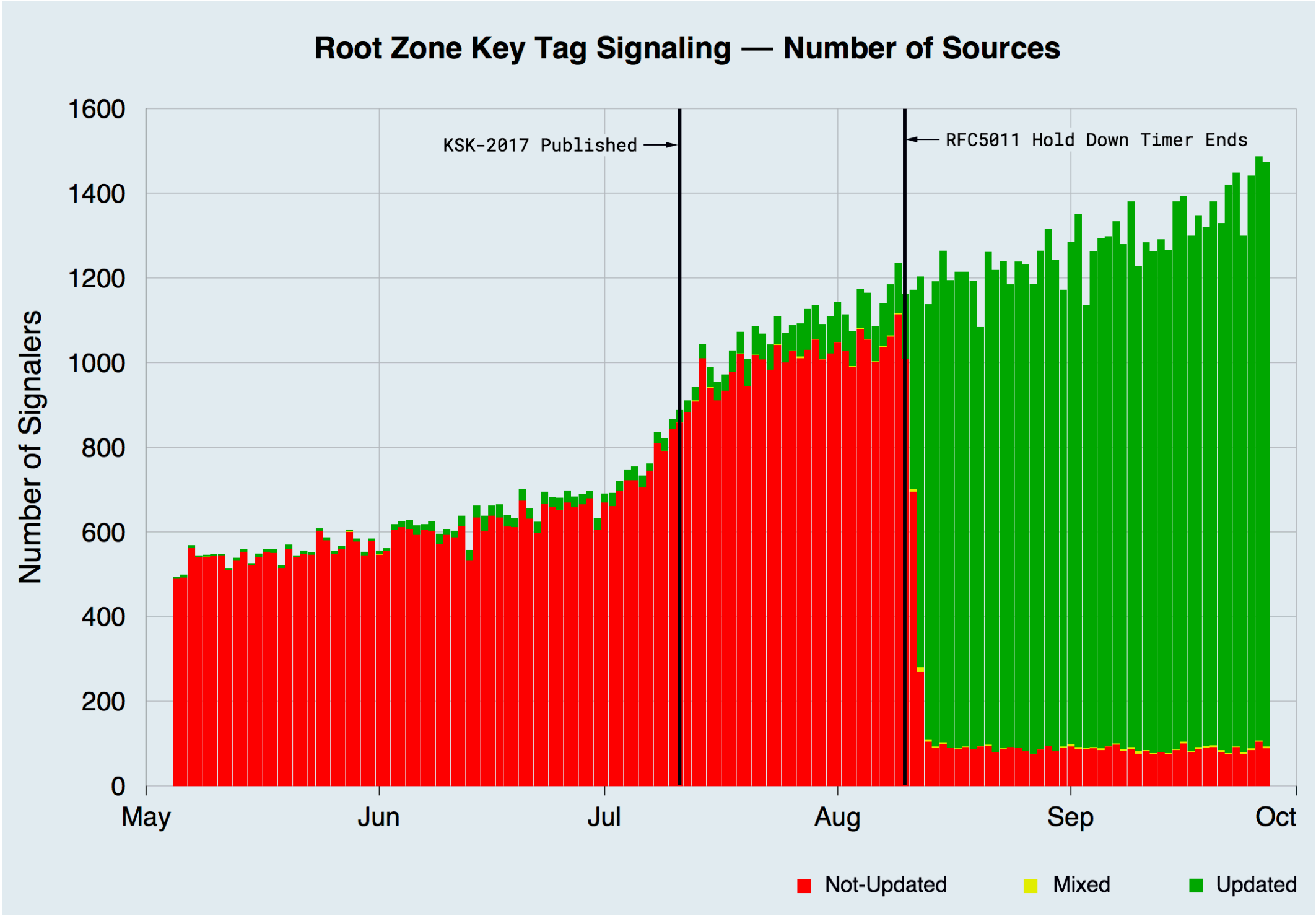
- October 27, 2016
  - KSK rollover process begins as the new KSK is generated.
- July 11, 2017
  - Publication of new KSK in DNS.
- September 19, 2017
  - Size increase for DNSKEY response from root name servers.
- October 11, 2017
  - New KSK begins to sign the root zone key set (the actual rollover event).

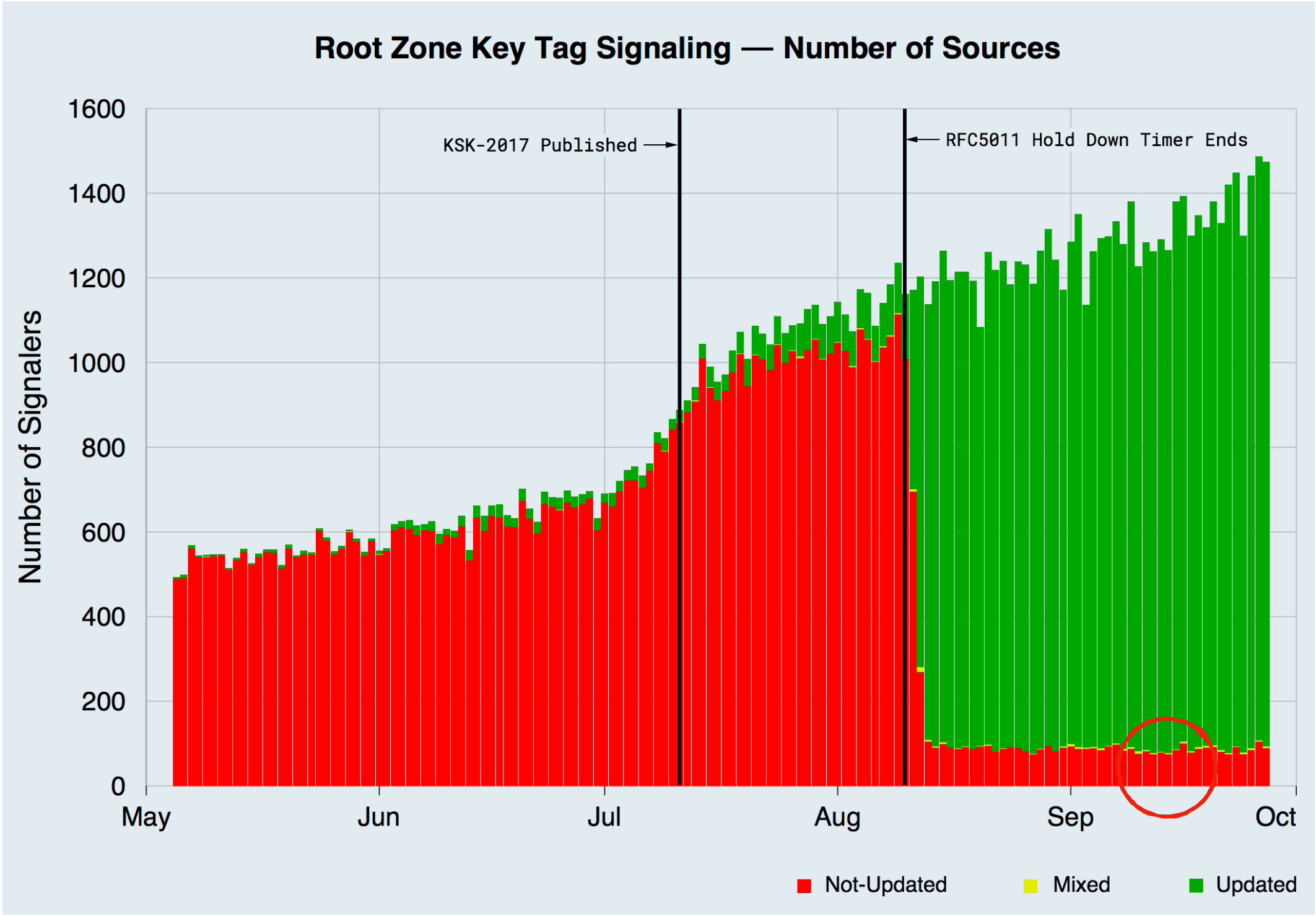
Root zone KSK rollover

- October 27, 2016
  - KSK rollover process begins as the new KSK is generated.
- July 11, 2017
  - Publication of new KSK in DNS.
- September 19, 2017
  - Size increase for DNSKEY response from root name servers.
- ~~**October 11, 2017**~~
  - ~~**New KSK begins to sign the root zone key set (the actual rollover event).**~~

What happened?

- The idea is that RFC 5011 automatic key rollover is used
  - Exceptions would be to update trust anchor manually
  - Massive information campaign have taken place
- RFC 8145 (*Signaling Trust Anchor Knowledge in DNSSEC*) is a way for a resolver to signal what trust anchors it has installed.
  - Implemented in Bind since mid-2016
  - Implemented in Unbound since mid-2017
- This data was collected after the keys where published in Sep 2017

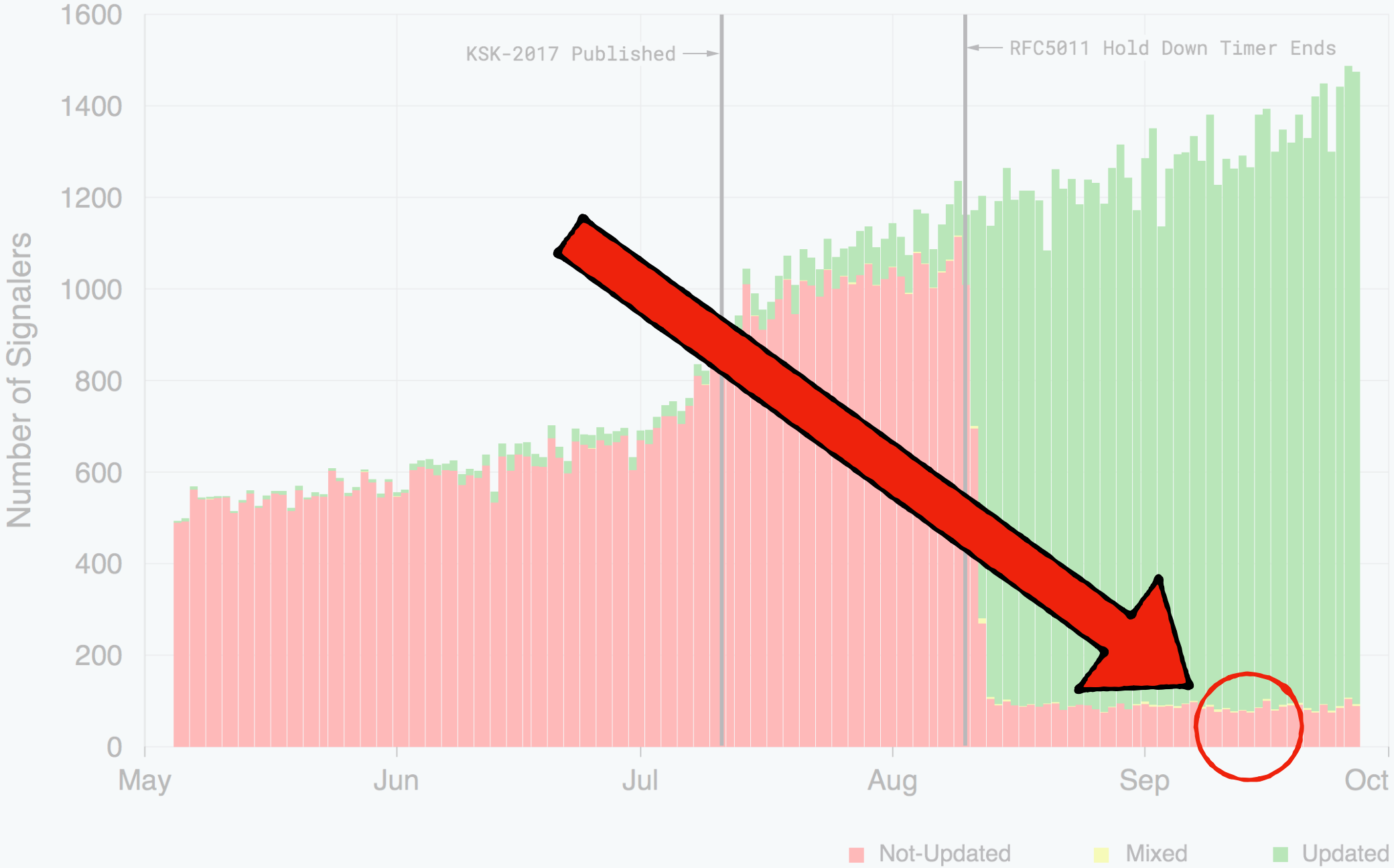

Root Zone Key Tag Signaling — Number of Sources
Number of Signalers
1600
1400
1200
1000
800
600
400
200
0
May
Jun
Jul
Aug
Sep
Oct
KSK-2017 Published
RFC5011 Hold Down Timer Ends
Not-Updated
Mixed
Updated

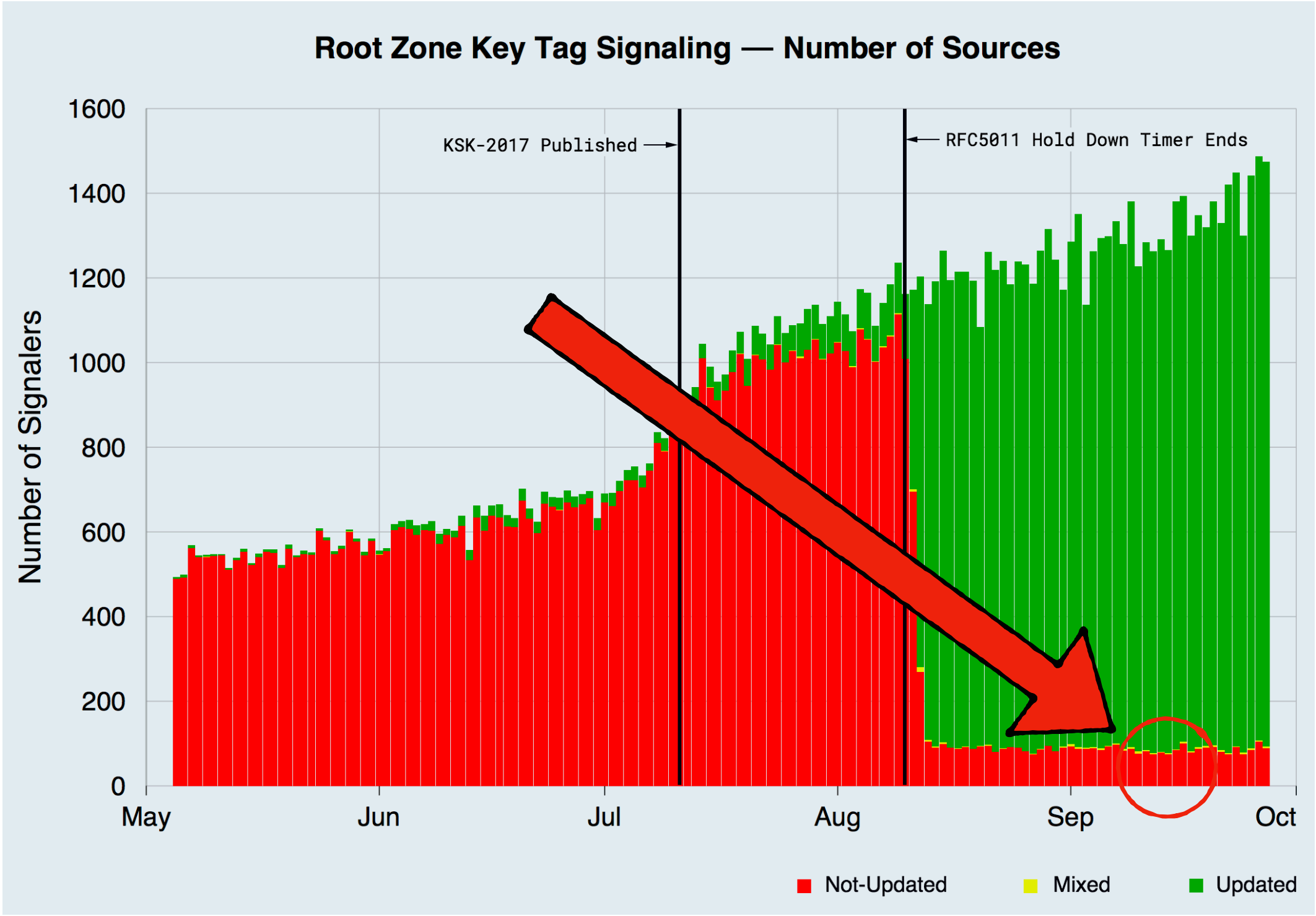

Root Zone Key Tag Signaling — Number of Sources
Number of Signalers
1600
1400
1200
1000
800
600
400
200
0
May
Jun
Jul
Aug
Sep
Oct
KSK-2017 Published
RFC5011 Hold Down Timer Ends
Not-Updated
Mixed
Updated

netnod

Root Zone Key Tag Signaling — Number of Sources

www.netnod.se




Root Zone Key Tag Signaling — Number of Sources
Number of Signalers
1600
1400
1200
1000
800
600
400
200
0
May
Jun
Jul
Aug
Sep
Oct
KSK-2017 Published
RFC5011 Hold Down Timer Ends
Not-Updated
Mixed
Updated

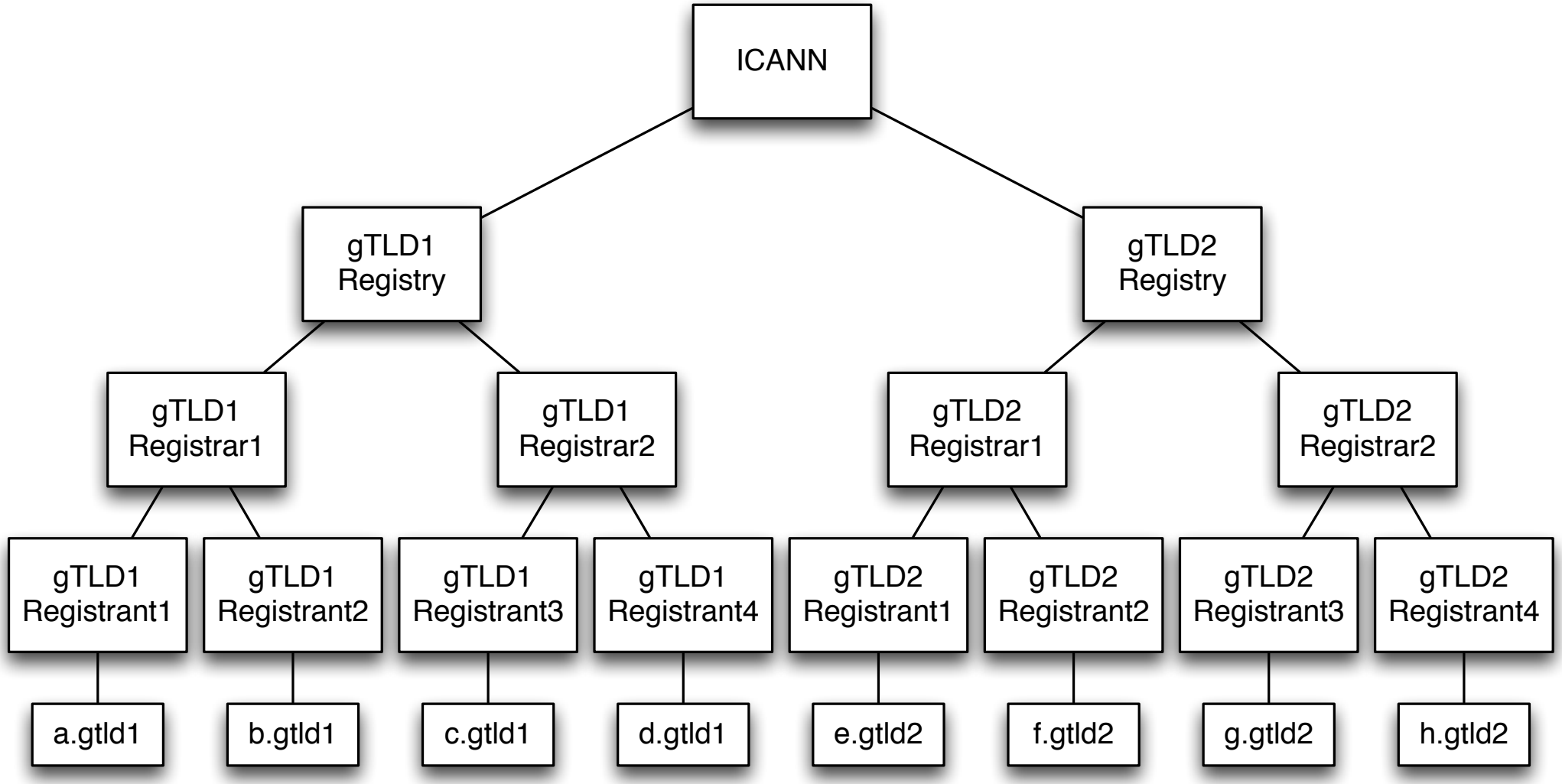
What does this mean?

What does this mean?

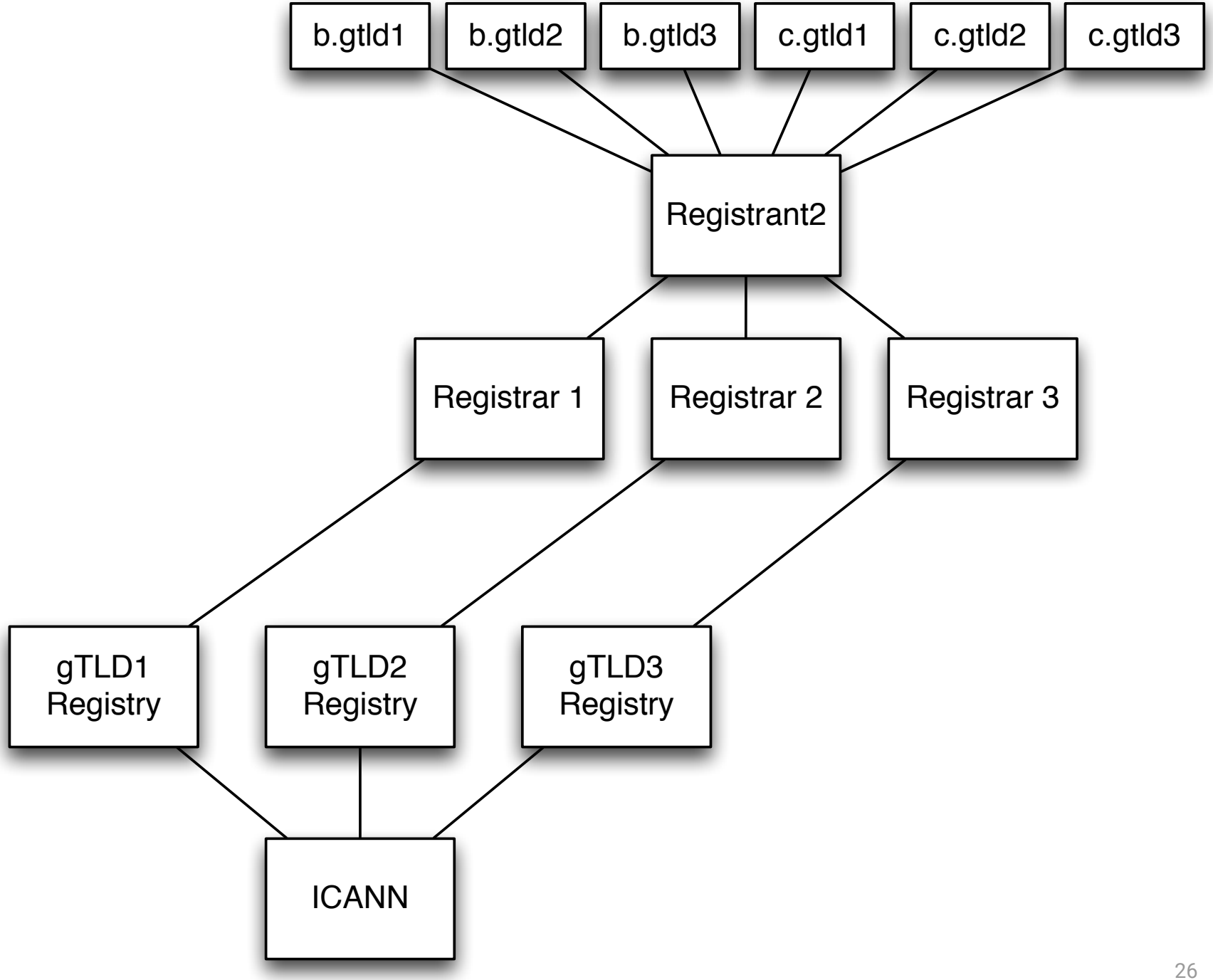
- We do not know!
- It could imply 5% of resolvers will fail DNSSEC validation
- It could imply 5% of resolvers have not fetched the right key
- It could imply 5% of deployed software have bugs
- We must evaluate the signal…

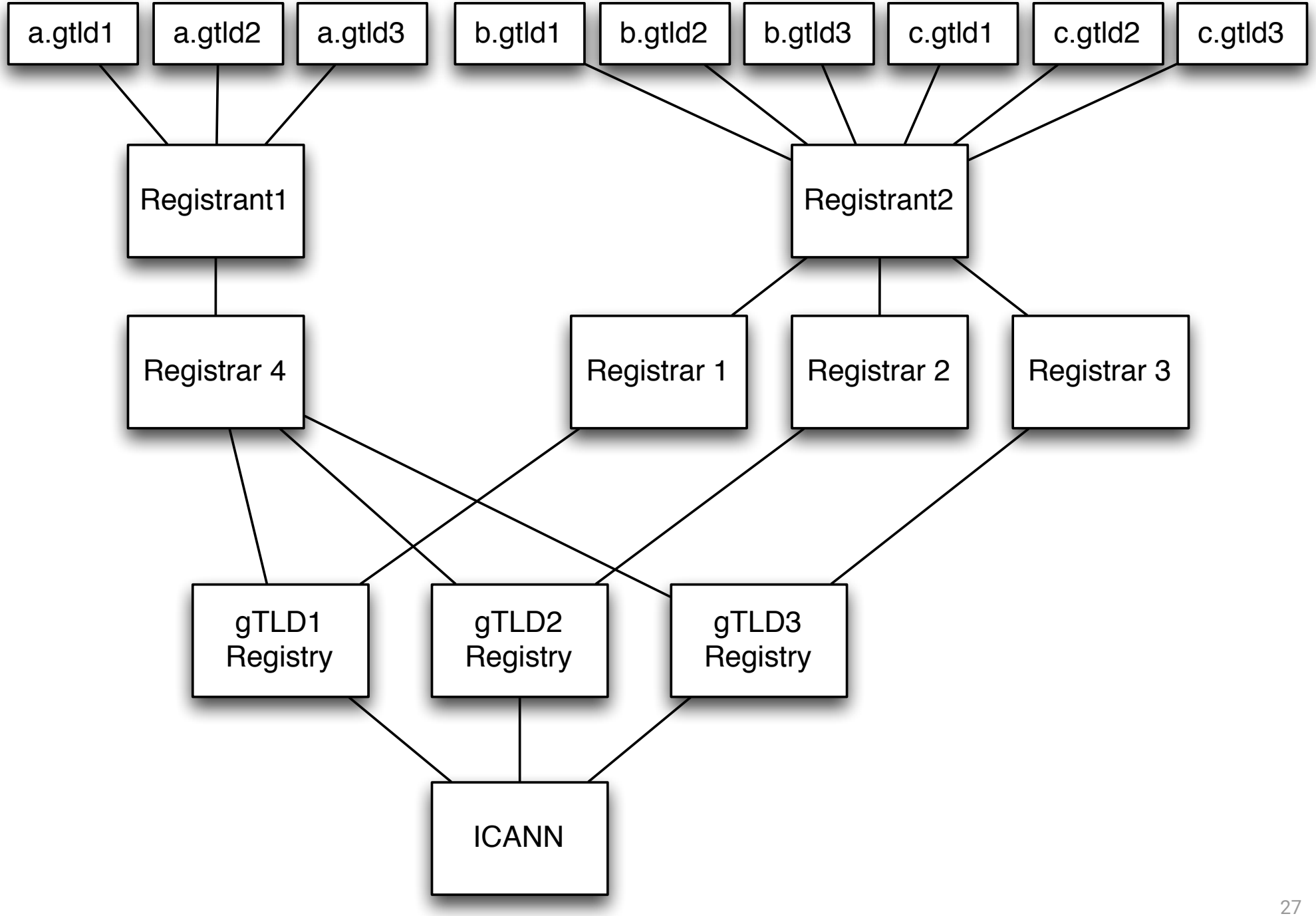
What should you do?

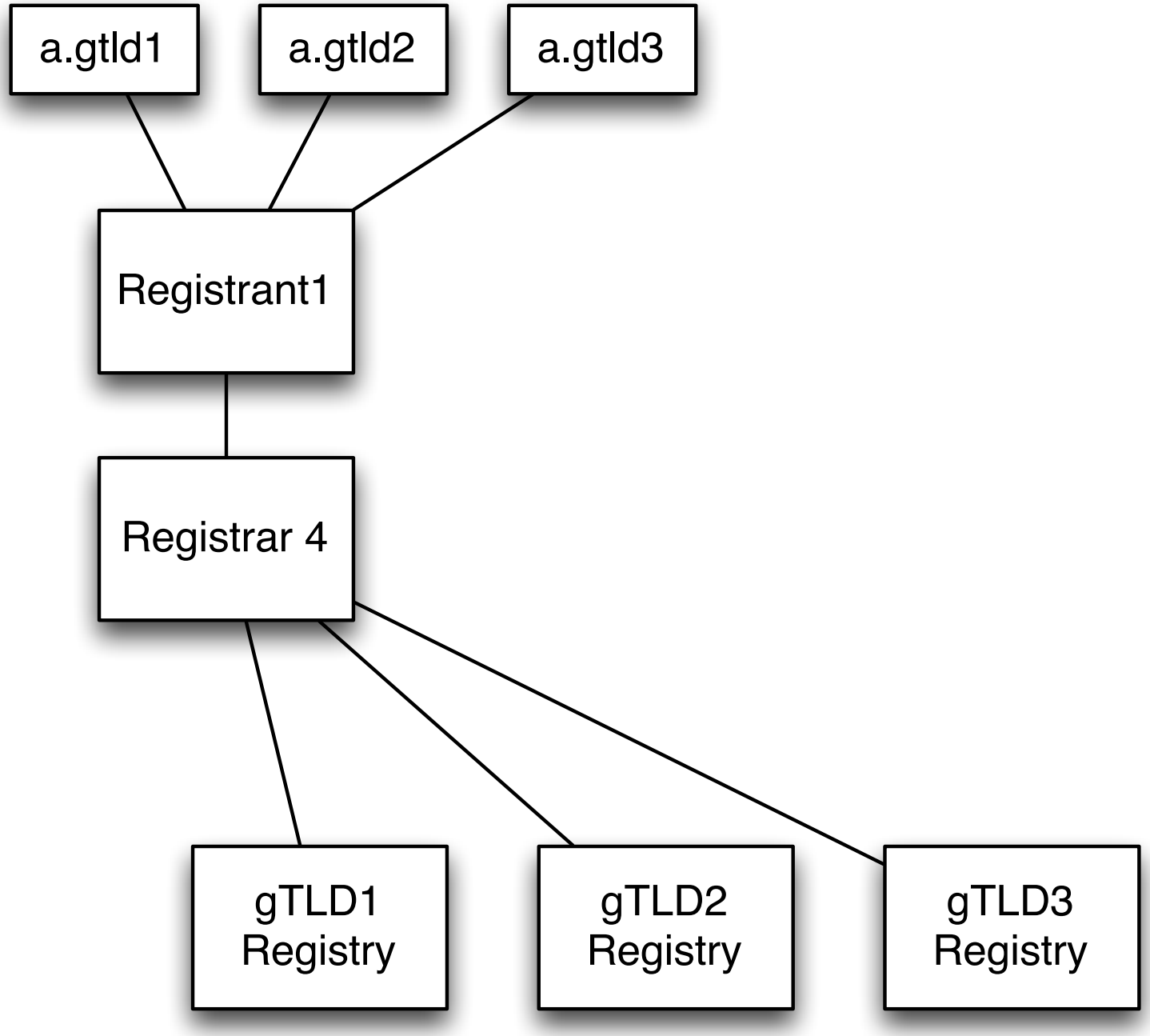
- Check to see you have all KSK’s in your resolver
  - https://www.icann.org/dns-resolvers-checking-current-trust-anchors
- Unbound:
  - auto-trust-anchor-file: ”/etc/unbound/root-anchors.key"
- Bind:
  - dnssec-validation auto;

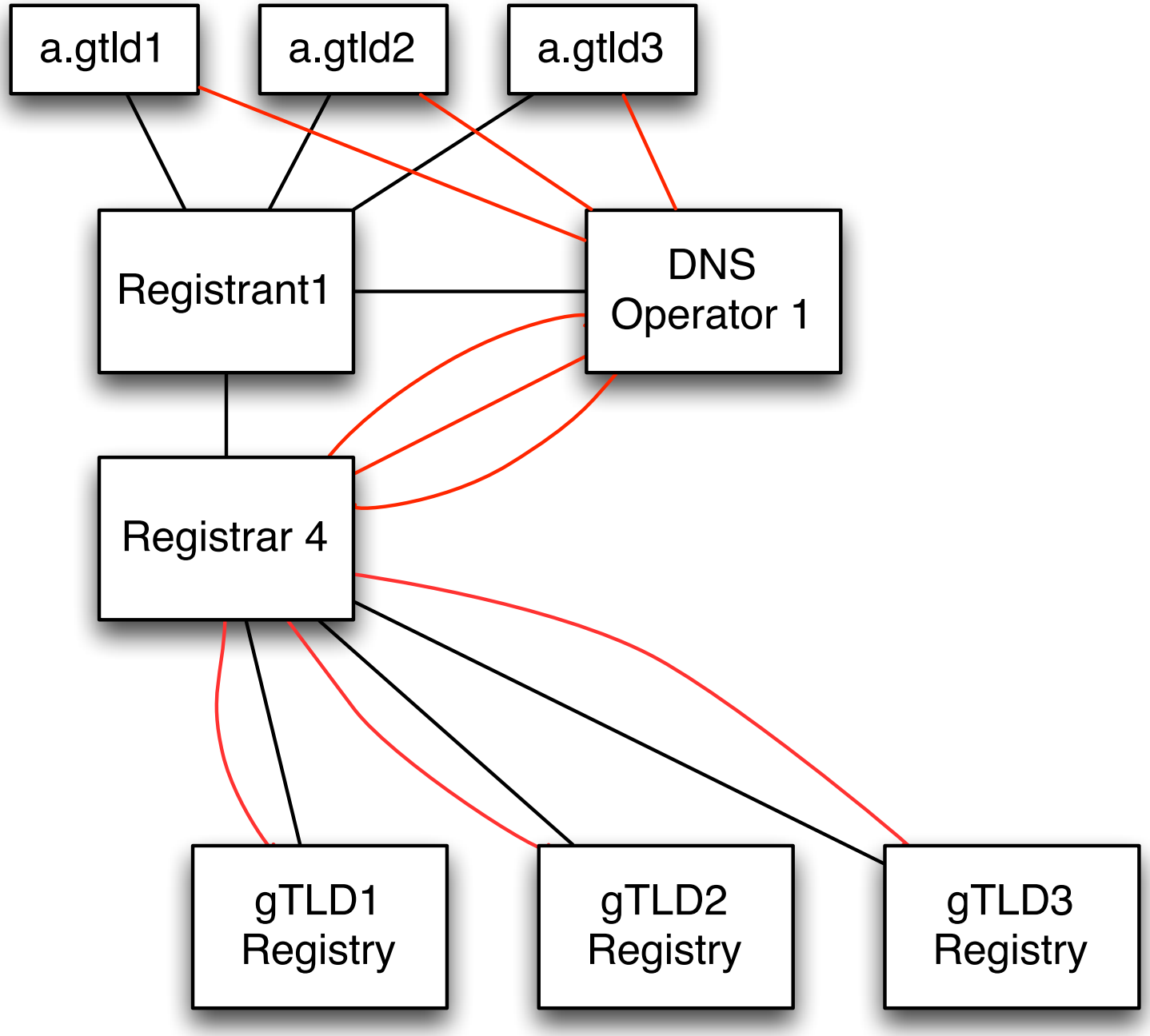

ICANN
gTLD1 Registry
gTLD2 Registry
gTLD1 Registrar1
gTLD1 Registrar2
gTLD2 Registrar1
gTLD2 Registrar2
gTLD1 Registrant1
gTLD1 Registrant2
gTLD1 Registrant3
gTLD1 Registrant4
gTLD2 Registrant1
gTLD2 Registrant2
gTLD2 Registrant3
gTLD2 Registrant4
a.gtld1
b.gtld1
c.gtld1
d.gtld1
e.gtld2
f.gtld2
g.gtld2
h.gtld2

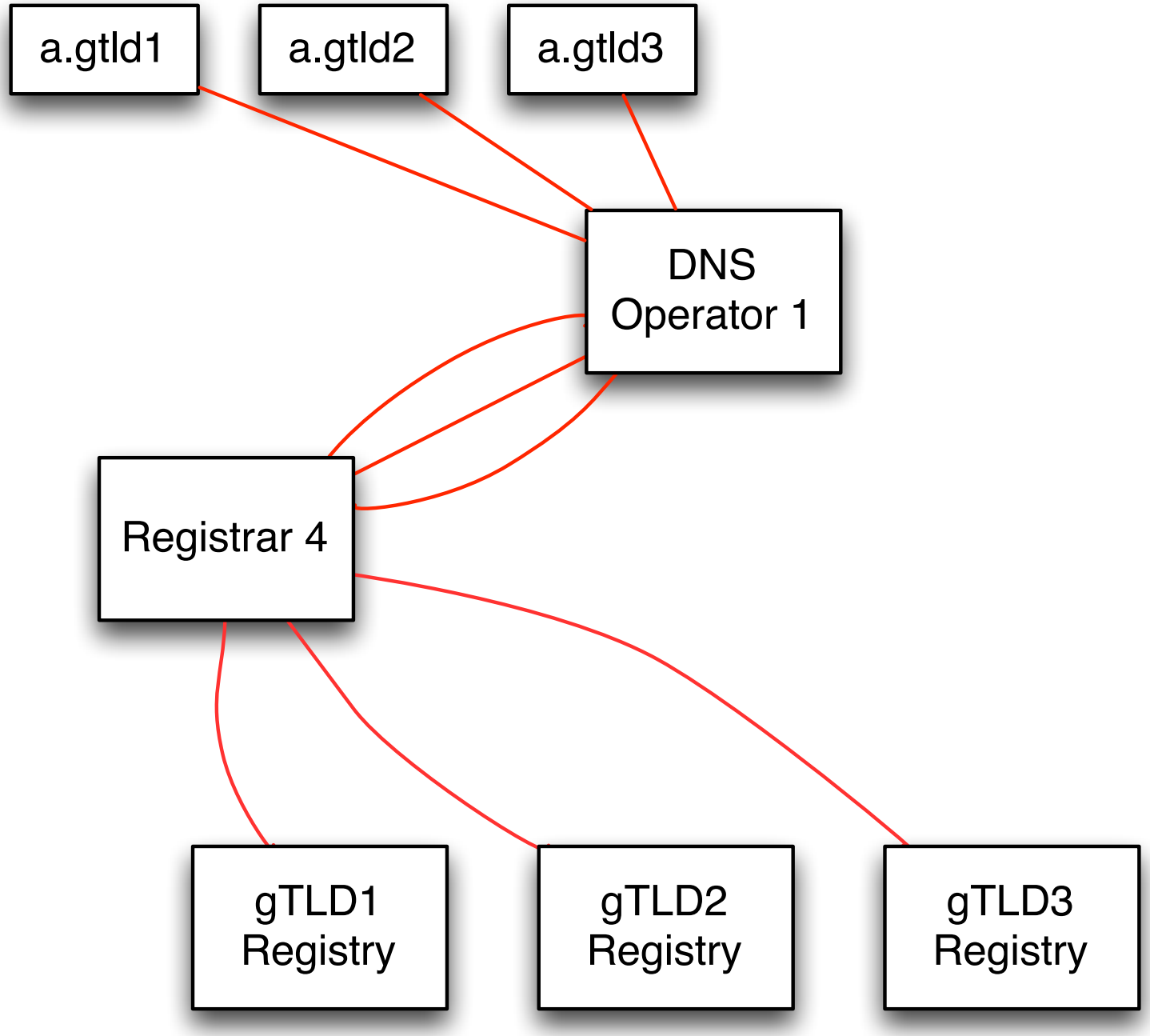
ICANN
gTLD1 Registry
gTLD2 Registry
gTLD1 Registrar1
gTLD1 Registrar2
gTLD2 Registrar1
gTLD2 Registrar2
gTLD1 Registrant1
gTLD1 Registrant2
gTLD1 Registrant3
gTLD1 Registrant4
gTLD2 Registrant1
gTLD2 Registrant2
gTLD2 Registrant3
gTLD2 Registrant4
a.gtld1
b.gtld1
c.gtld1
d.gtld1
e.gtld2
f.gtld2
g.gtld2
h.gtld2

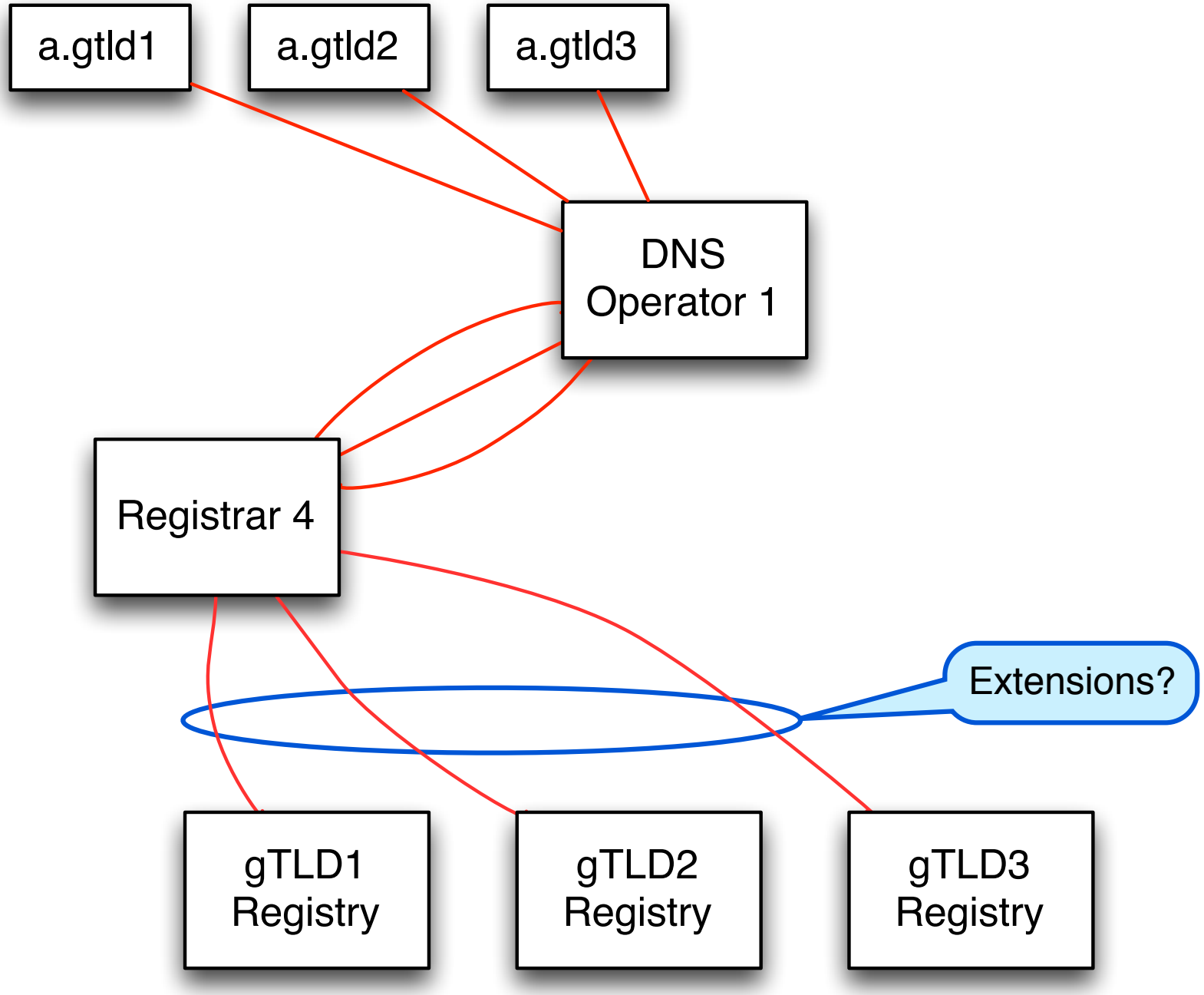

b.gtld1
b.gtld2
b.gtld3
c.gtld1
c.gtld2
c.gtld3
Registrant2
Registrar 1
Registrar 2
Registrar 3
gTLD1 Registry
gTLD2 Registry
gTLD3 Registry
ICANN


a.gtld1
a.gtld2
a.gtld3
b.gtld1
b.gtld2
b.gtld3
c.gtld1
c.gtld2
c.gtld3
Registrant1
Registrant2
Registrar 4
Registrar 1
Registrar 2
Registrar 3
gTLD1 Registry
gTLD2 Registry
gTLD3 Registry
ICANN


a.gtld1
a.gtld2
a.gtld3
Registrant1
Registrar 4
gTLD1
Registry
gTLD2
Registry
gTLD3
Registry


a.gtld1
a.gtld2
a.gtld3
Registrant1
DNS
Operator 1
Registrar 4
gTLD1
Registry
gTLD2
Registry
gTLD3
Registry


a.gtld1
a.gtld2
a.gtld3
DNS Operator 1
Registrar 4
gTLD1 Registry
gTLD2 Registry
gTLD3 Registry


a.gtld1
a.gtld2
a.gtld3
DNS Operator 1
Registrar 4
Extensions?
gTLD1 Registry
gTLD2 Registry
gTLD3 Registry

Perl Library Net::DRI version 0.96

Net::DRI::DRD - Superclass of all Net::DRI Registry Drivers

48 different: ASIA, AT, AU, AdamsNames, BE, BIZ, BR, BZ, CAT, CIRA, COOP, CZ, CoCCA, DENIC, EURid, GL, Gandi, HN, ICANN, IENUMAT, IM, INFO, IRegistry, IT, LC, LU, ME, MN, MOBI, NAME, NO, NU, Nominet, ORG, OVH, OpenSRS, PL, PRO, PT, SC, SE, SIDN, SWITCH, TRAVEL, US, VC, VNDS and WS

Net::DRI::Protocol::EPP::Extensions - Various extensions

34 different: AERO, AFNIC, ARNES, ASIA, AT, AU, Afilias, BR, CAT, CIRA, COOP, CZ, CentralNic, DNSBE, EurID, FCCN, GracePeriod, IENUMAT, IRegistry, IT, LU, MOBI, NAME, NO, NSgroup, NeuLevel, Nominet, PL, PRO, SE, SIDN, SWITCH, US and VeriSign

Specifically for DNSSEC

```
5910 Domain Name System (DNS) Security Extensions Mapping for the
     Extensible Provisioning Protocol (EPP). J. Gould, S. Hollenbeck. May
     2010. (Format: TXT=72490 bytes) (Obsoletes RFC4310) (Status: PROPOSED
     STANDARD)
```

Specifically for DNSSEC

4. DS Data Interface and Key Data Interface

This document describes operational scenarios in which a client can create, add, and remove Delegation Signer (DS) information or key data information for a domain name. There are two different forms of interfaces that a server can support. The first is called the "DS Data Interface", where the client is responsible for the creation of the DS information and is required to pass DS information when performing adds and removes. The server is required to pass DS information for <domain:info> responses. The second is the "Key Data Interface," where the client is responsible for passing the key data information when performing adds and removes. The server is responsible for passing key data information for <domain:info> responses.

Specifically for DNSSEC

4. DS Data Interface and Key Data Interface

This document describes operational scenarios in which a client can create, add, and remove Delegation Signer (DS) information or key data information for a domain name. **There are two different forms of interfaces that a server can support. The first is called the "DS Data Interface"**, where the client is responsible for the creation of the DS information and is required to pass DS information when performing adds and removes. The server is required to pass DS information for <domain:info> responses. **The second is the "Key Data Interface,"** where the client is responsible for passing the key data information when performing adds and removes. The server is responsible for passing key data information for <domain:info> responses.

Why two?

Search for ***accepting DS vs DNSKEY*** and you find discussions that have been going on for as long as we have been discussing DNSSEC, registry/registrar model and epp.

We have not been able to converge...

Until now!

- RFC7344 - Automating DNSSEC Delegation Trust Maintenance
  - CDNSKEY/CDS Records
- RFC 8078 - Managing DS Records from the Parent via CDS/CDNSKEY
  - Enable DNSSEC
  - Rollover DNSKEY
  - Disable DNSSEC

DLV at ISC

- Have existed since 2006
  - When ISC created the DLV Registry, we were years away from the .com/.net/.org zones (let alone the root zone) from being signed, so there was a chicken and egg problem
  - Organizations wouldn't deploy DNSSEC because there was no easy way to have a validated chain of trust
  - DLV allowed organizations to sign their zones, and the caching resolver (if it has DLV validation support enabled) to validate the keys
- Bootstrap mechanism until the root zone was signed
  - Which happened a number of years ago…
- It is finally closed!
  - https://www.isc.org/blogs/dlv/

PATRIK FÄLTSTRÖM

Head of Technology
Netnod

paf@netnod.se