DNSSEC
ROLLING KEYS

Presented by
Olaf Kolkman (NLnet Labs)

NLnet Labs

DNSKEY in flavours

- Zone Signin Key (ZSK)
- Key Signing Key (KSK)
  - Functions as secure entry point into the zone
    - Trust-anchor configuration
    - Parental DS points to it
    - Interaction with 3rd party
- DNSKEYs are treated all the same in the protocol
- Operators can make a distinction
  - Look at the flag field: ODD (257 in practice) means SEP

NLnet Labs

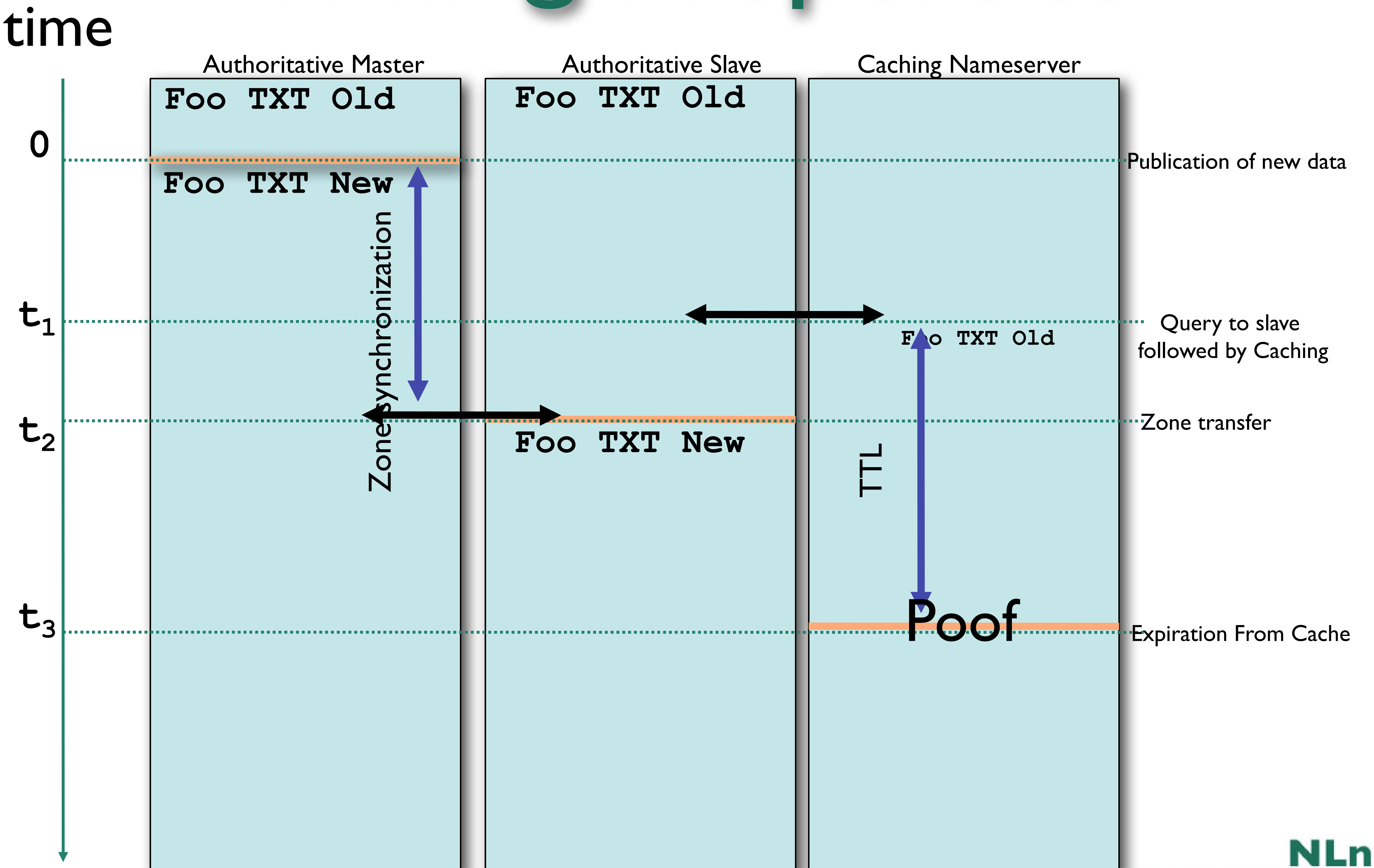
Benefits of using separate keys

- Rolling KSK needs interaction, rolling ZSKs can be done almost instantaneously
- Remember KSK replacement may result in
  - Trust-anchor updates
  - Change of DS record at parent
- Allows different responsibilities
  - ZSKs may be touched day to day by junior staff
  - KSKs may only be touched by senior staff

Rolling keys instantaneously?

- Remember that in the DNS caches are at play.
  - It takes a bit of time to have new information propagate
- When you happen to get new DNSKEYs you would like to be able to use DNSSIGs from the cache
- When you happen to get old DNSKEYs from the cache you would like to use new DNSSIGs
- Try to make sure both old and new keys are available
- Or, try to make sure both old and new sigs are available

Timing Properties

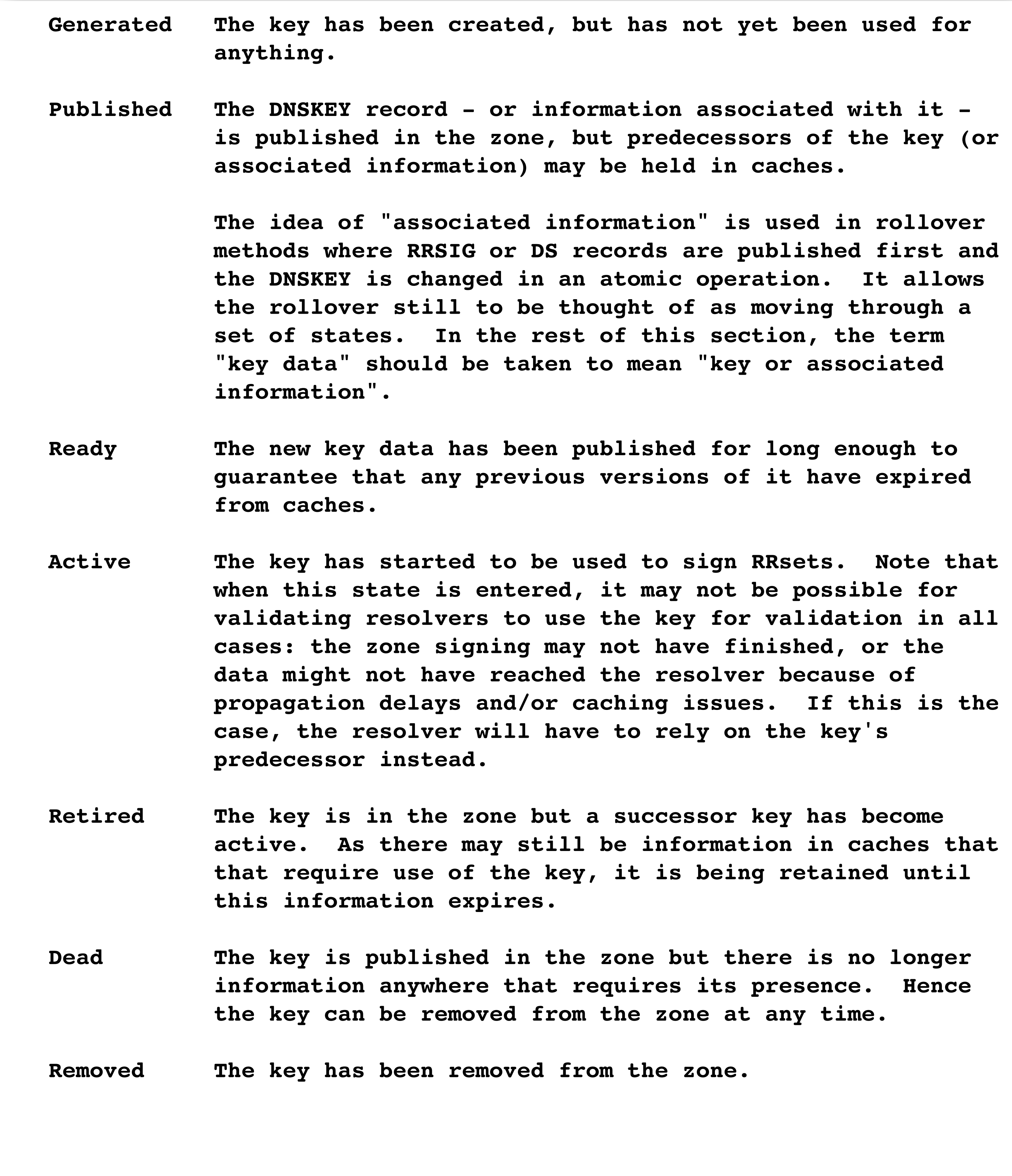


PRE-publish ZSK rollover

- Introduce the new DNSKEY before you start using it to sign the data.
  - ‘published and active’ key
  - The published key is just published, the active key is used for signing
- You could also create two signatures after introducing the key, but that would cause your zone file to grow

For those that write scripts/programs

draft-ietf-dnsop-dnssec-key-timing provides a definition of the various states and transitions you need to take into account



Generated   The key has been created, but has not yet been used for anything.

Published   The DNSKEY record - or information associated with it - is published in the zone, but predecessors of the key (or associated information) may be held in caches.

The idea of "associated information" is used in rollover methods where RRSIG or DS records are published first and the DNSKEY is changed in an atomic operation. It allows the rollover still to be thought of as moving through a set of states. In the rest of this section, the term "key data" should be taken to mean "key or associated information".

Ready   The new key data has been published for long enough to guarantee that any previous versions of it have expired from caches.

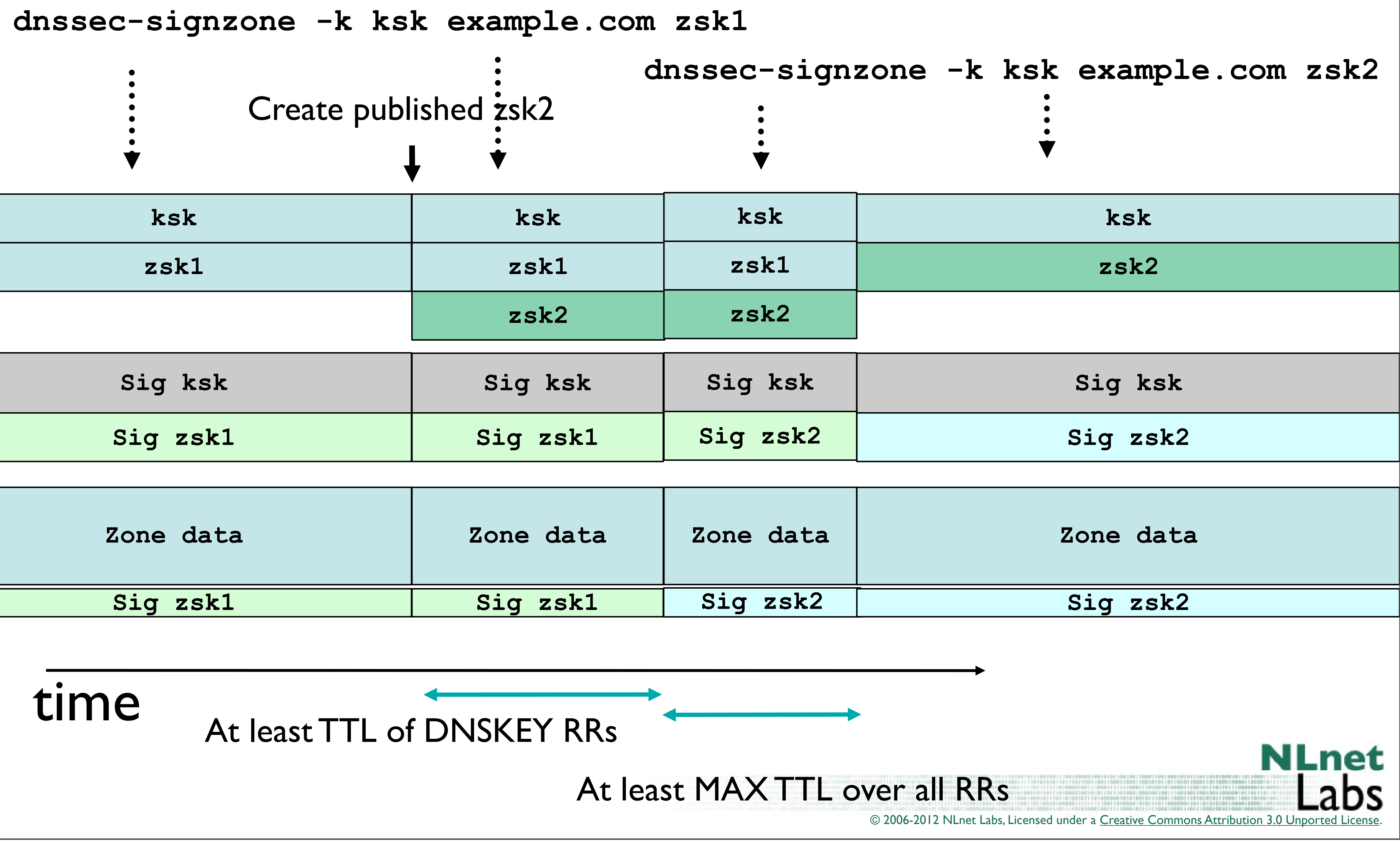
Active   The key has started to be used to sign RRsets. Note that when this state is entered, it may not be possible for validating resolvers to use the key for validation in all cases: the zone signing may not have finished, or the data might not have reached the resolver because of propagation delays and/or caching issues. If this is the case, the resolver will have to rely on the key's predecessor instead.

Retired   The key is in the zone but a successor key has become active. As there may still be information in caches that that require use of the key, it is being retained until this information expires.

Dead   The key is published in the zone but there is no longer information anywhere that requires its presence. Hence the key can be removed from the zone at any time.

Removed   The key has been removed from the zone.

ZSK rollover



```
dnssec-signzone -k ksk example.com zsk1
```

Create published zsk2

```
dnssec-signzone -k ksk example.com zsk2
```

ksk	ksk	ksk	ksk
zsk1	zsk1	zsk1	zsk2
	zsk2	zsk2	
Sig ksk	Sig ksk	Sig ksk	Sig ksk
Sig zsk1	Sig zsk1	Sig zsk2	Sig zsk2
Zone data	Zone data	Zone data	Zone data
Sig zsk1	Sig zsk1	Sig zsk2	Sig zsk2

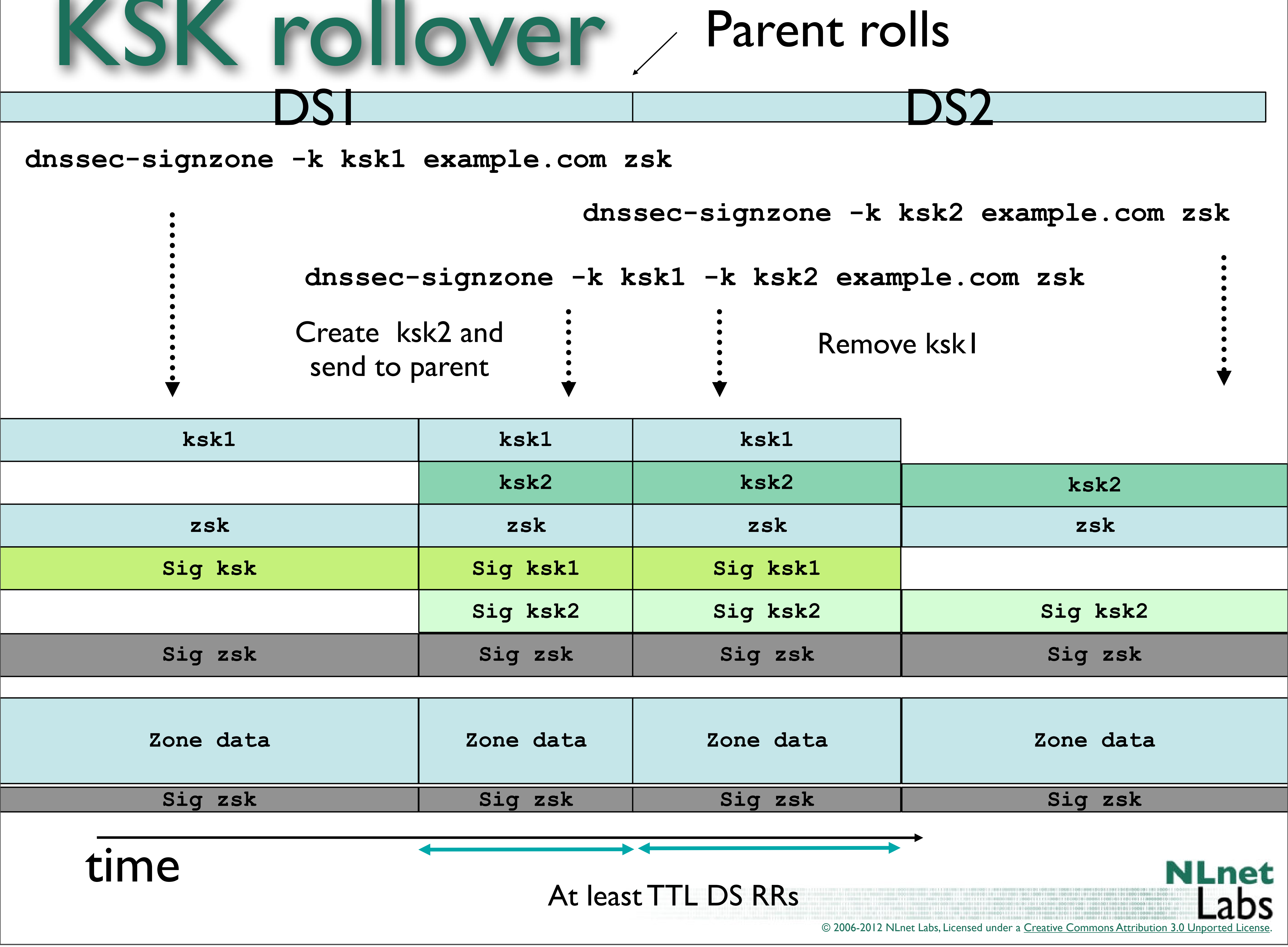
At least TTL of DNSKEY RRs

At least MAX TTL over all RRs

KSK rollover

- You are dependent on your parent.
  - You cannot control when the parent changes the DS rr
- Use the old KSK until the old DNS had time to propagate from caches

KSK rollover



Parent rolls

DS1

DS2

```
dnssec-signzone -k ksk1 example.com zsk
```

```
dnssec-signzone -k ksk2 example.com zsk
```

```
dnssec-signzone -k ksk1 -k ksk2 example.com zsk
```

Create ksk2 and send to parent

Remove ksk1

ksk1	ksk1	ksk1	
	ksk2	ksk2	ksk2
zsk	zsk	zsk	zsk
Sig ksk	Sig ksk1	Sig ksk1	
	Sig ksk2	Sig ksk2	Sig ksk2
Sig zsk	Sig zsk	Sig zsk	Sig zsk
Zone data	Zone data	Zone data	Zone data
Sig zsk	Sig zsk	Sig zsk	Sig zsk

time

At least TTL DS RRs

Erratum

```
----------------------------------------------------------------
initial         new DNSKEY        new RRSIGs      DNSKEY removal
----------------------------------------------------------------
SOA0            SOA1              SOA2            SOA3
RRSIG10(SOA0)   RRSIG10(SOA1)     RRSIG11(SOA2)   RRSIG11(SOA3)

DNSKEY1         DNSKEY1           DNSKEY1         DNSKEY1
DNSKEY10        DNSKEY10          DNSKEY10        DNSKEY11
DNSKEY11        DNSKEY11
RRSIG1 (DNSKEY) RRSIG1 (DNSKEY)   RRSIG1(DNSKEY)  RRSIG1 (DNSKEY)
RRSIG10(DNSKEY) RRSIG10(DNSKEY)   RRSIG11(DNSKEY) RRSIG11(DNSKEY)
----------------------------------------------------------------

                     Pre-Publish Key Rollover
```

Erratum II

```
----------------------------------------------------------------
initial             new DNSKEY         DNSKEY removal
----------------------------------------------------------------
SOA0                SOA1               SOA2
RRSIG10(SOA0)       RRSIG10(SOA1)      RRSIG11(SOA2)
RRSIG11(SOA1)

DNSKEY1             DNSKEY1            DNSKEY1
DNSKEY10            DNSKEY10           DNSKEY11
DNSKEY11
RRSIG1(DNSKEY)      RRSIG1(DNSKEY)     RRSIG1(DNSKEY)
RRSIG10(DNSKEY)     RRSIG10(DNSKEY)    RRSIG11(DNSKEY)
RRSIG11(DNSKEY)
----------------------------------------------------------------
       Double Signature Zone Signing Key Rollover
```

Key Management

- There are many keys to maintain
  - Keys are used on a per zone basis
    - Key Signing Keys and Zone Signing Keys
  - During key rollovers there are multiple keys
    - In order to maintain consistency with cached DNS data [RFC4641]
- Private keys need shielding

KSK and trust-anchors

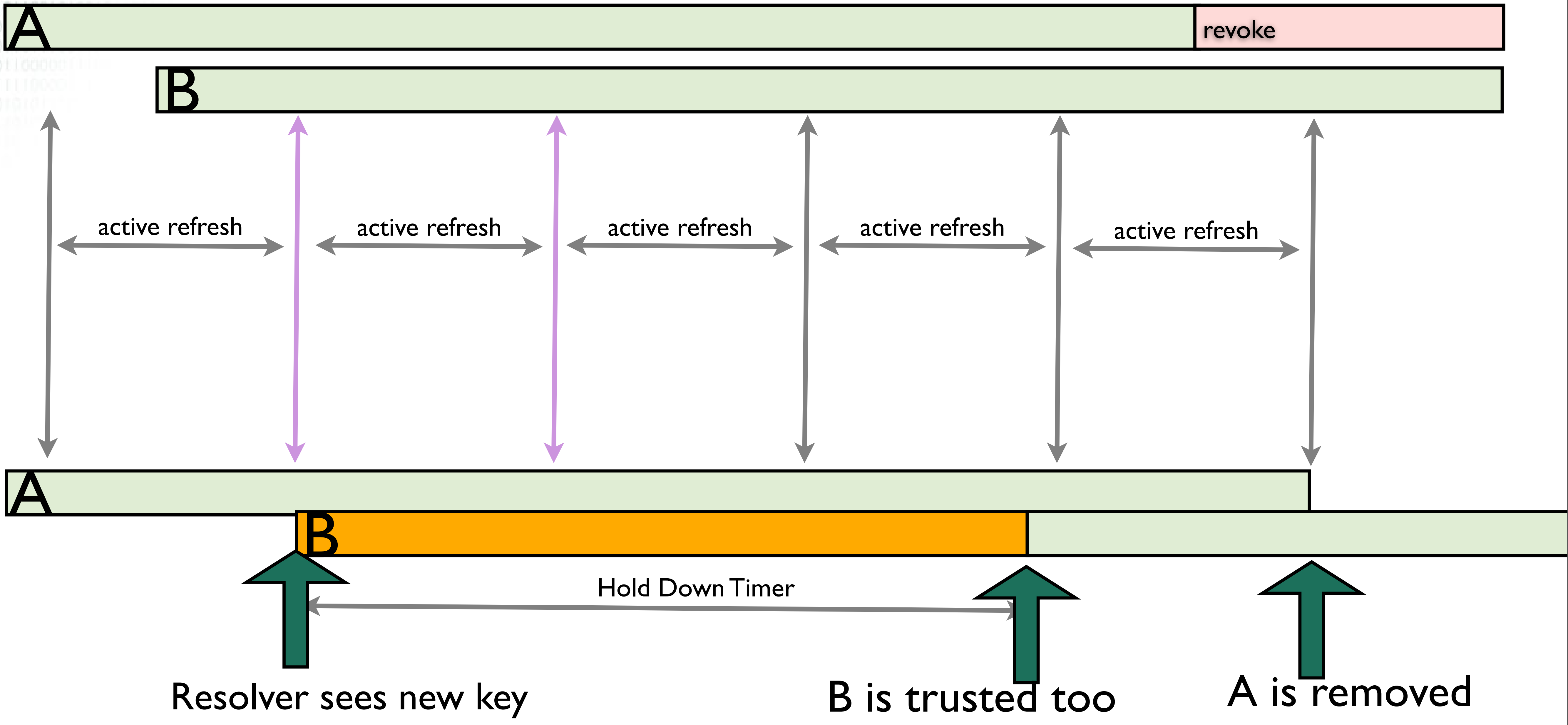
- If your KSK is used as a trust anchor (do you know?) you should deploy mechanisms to allow automated rollover
- RFC 5011

RFC 5011 Concepts

- Trust anchor maintenance based on existing trust relation
- New keys only accepted after its been seen for more than 30 days (Hold Down)
- Signaling retirement of the key by setting a ‘revoke’ flag

NLnet Labs

Zone Owner

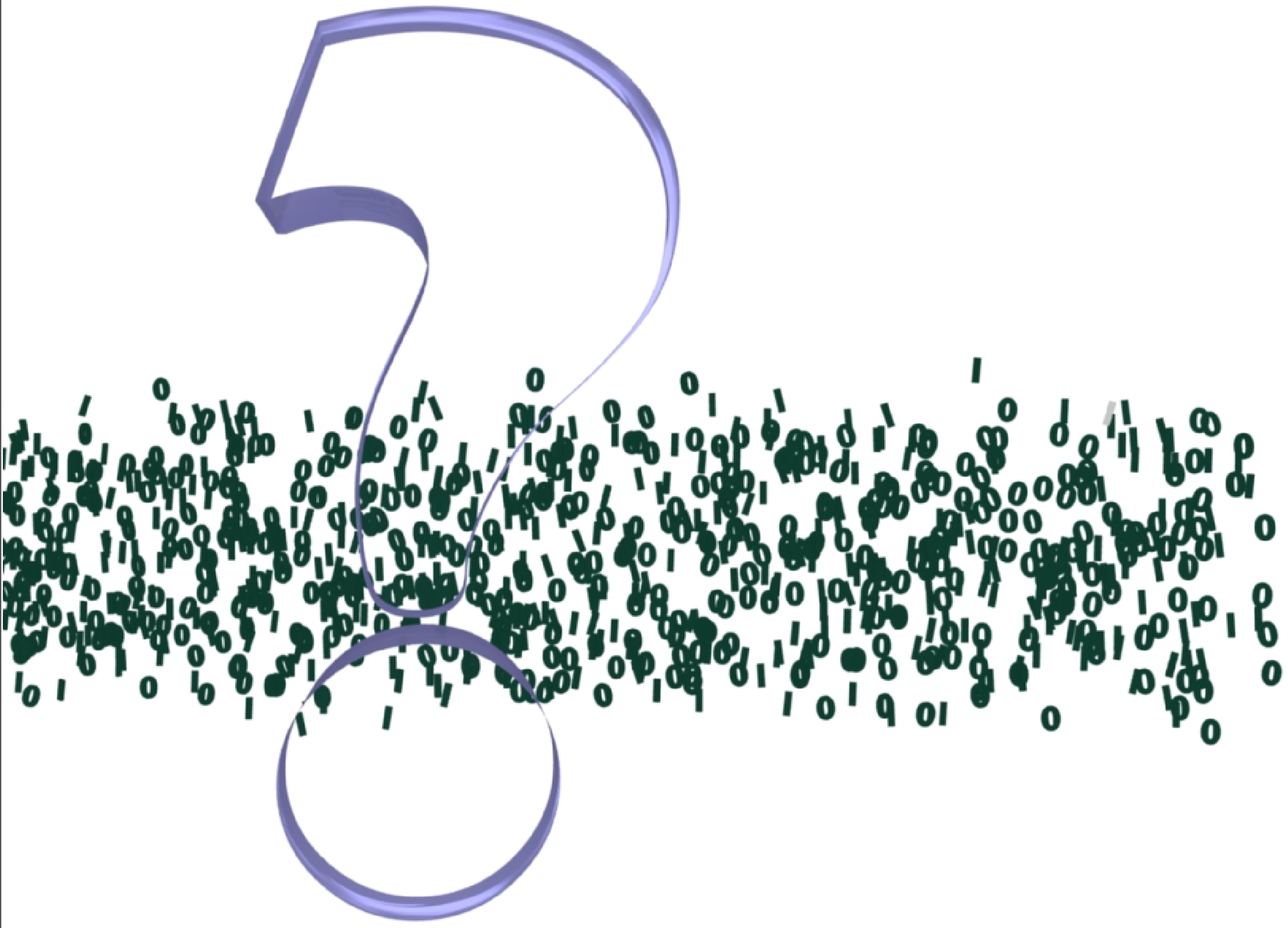


Trust Anchor state

By Hand?

- You can do this using tools manually
- Error prone and complex
- Easy to forget
- Use tools

Key managment



- Not only a matter of changing a key
- You have to keep the DNS timing properties into account
- Manual, yes for only a few zones
- Error prone!!!