DNSSEC Deployment

Presented by

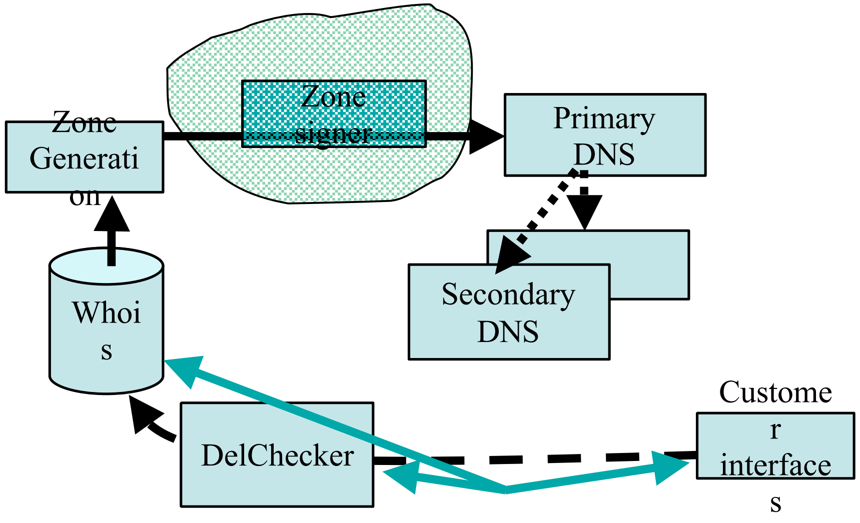
Olaf Kolkman (NLnet Labs)

and

Alain Aina(TRS)

Rabat Morocco, June 1, 2008

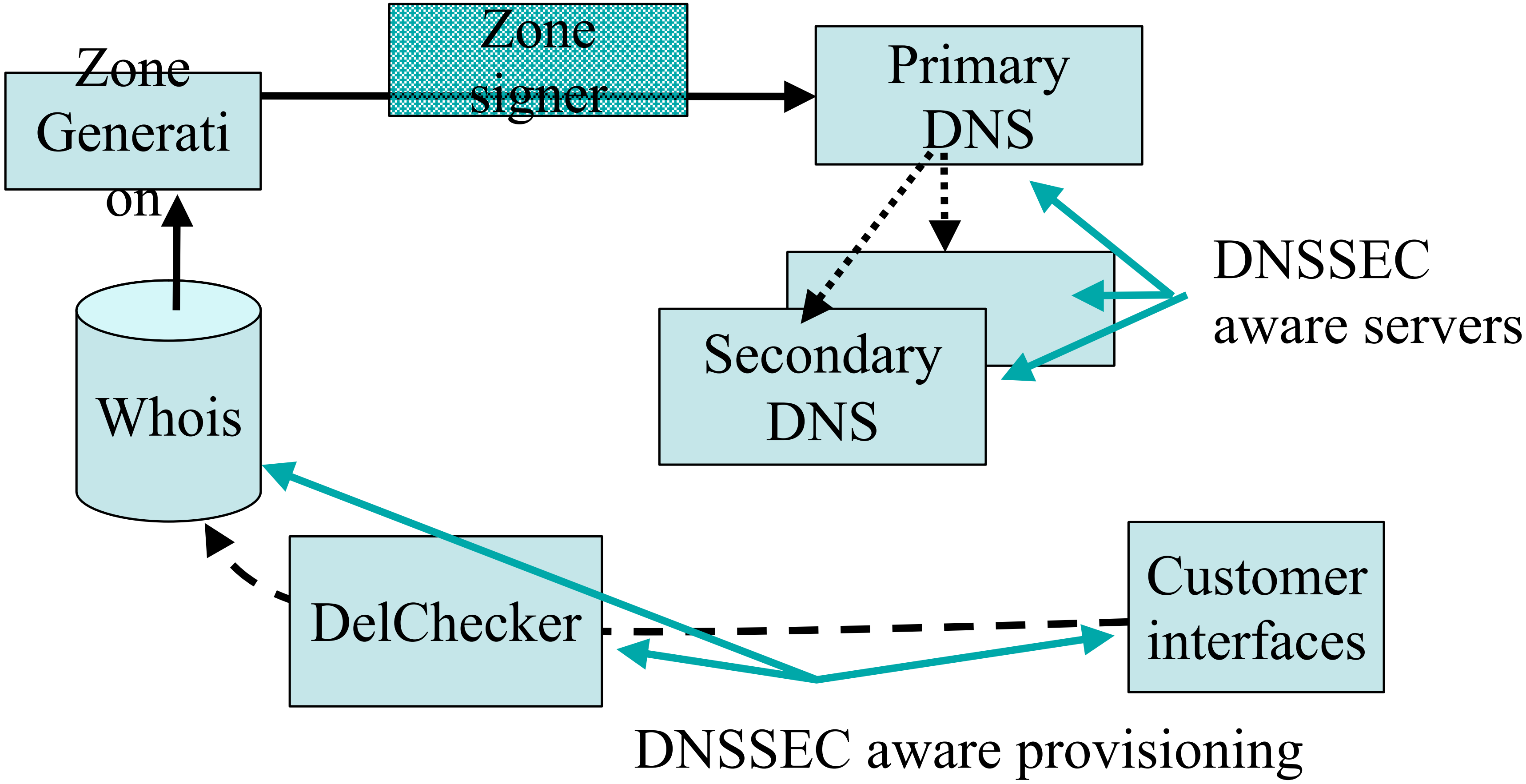
Presentation roadmap



DNSSEC aware provisioning

- Overview of problem space
  - Architectural changes to allow for DNSSEC deployment
- Deployment tasks
  - Key maintenance
  - DNS server infrastructure
  - Providing secure delegations

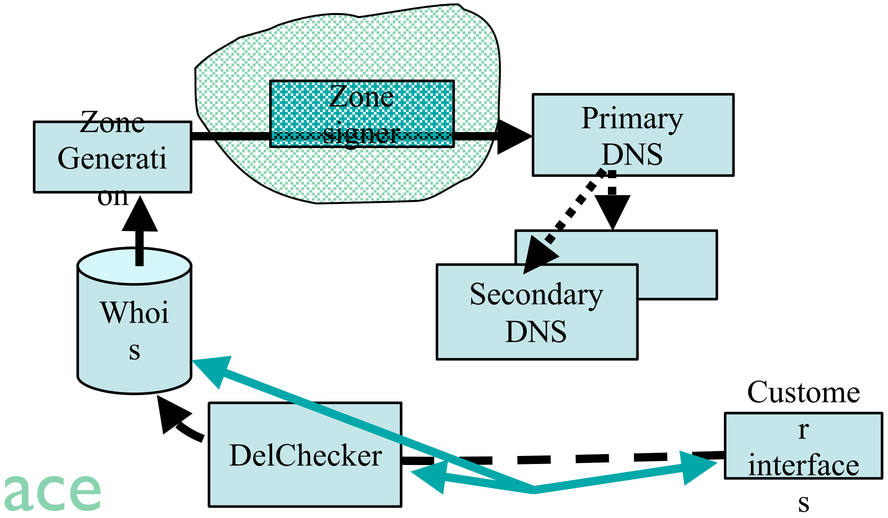
DNSSEC Architecture modifications



DNSSEC deployment tasks

- Key maintenance policies and tools
  - Private Key use and protection
  - Public key distribution
- Zone signing and integration into the provisioning chain
- DNS server infrastructure
- Secure delegation registry changes
  - Interfacing with customers

Presentation roadmap



DNSSEC aware provisioning

- Overview of problem space
  - Architectural changes to allow for DNSSEC deployment
- Deployment tasks
  - Key maintenance
  - DNS server infrastructure
  - Providing secure delegations

Key Maintenance

- DNSSEC is based on public key cryptography
  - Data is signed using a private key
  - It is validated using a public key

Operational problems:

- Dissemination of the public key
- Private key has a ‘*best before*’ date
  - Keys change, and the change has to disseminate

Public Key Dissemination

- In theory only one trust-anchor needed that of the root
  - How does the root key get to the end user?
  - How is it rolled?
- In absence of hierarchy there will be many trust-anchors
  - How do these get to the end-users?
  - How are these rolled?
- These are open questions, making early deployment difficult.

Public Key Dissemination at RIPE NCC

In absence of a signed parent zone and automatic rollover:

- Trust anchors are published on an “HTTPS” secured website
- Trust anchors are signed with the RIPE NCC public keys
- Trust anchor will be rolled twice a year (during early deployment)
- Announcements and publications are always signed by x.509 or PGP

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

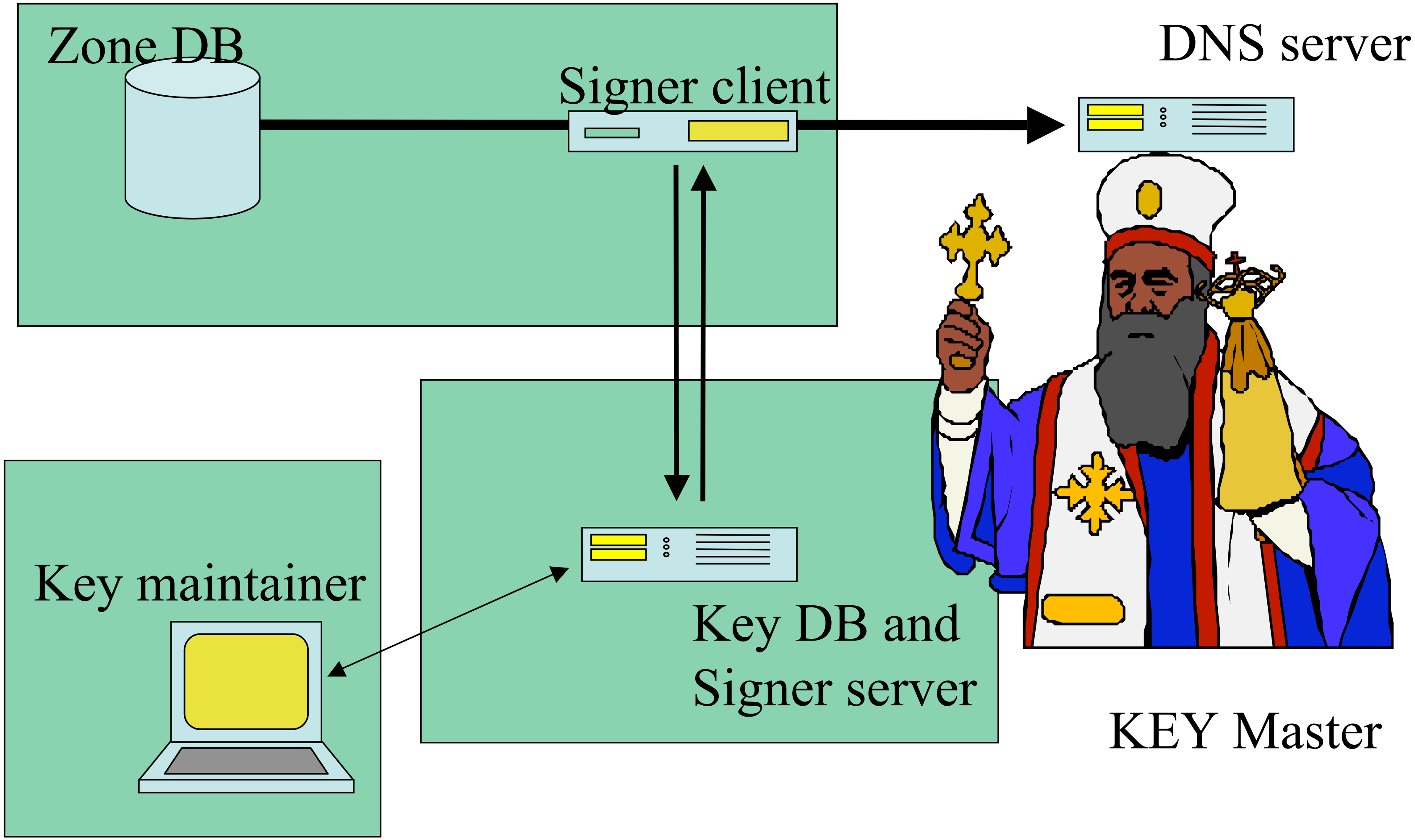
Approaches

- Use of a smart card to store the KSK
  - http://www.iis.se/pdf/dnssec-techenv-en.pdf
- The use of hardware signers and management software
  - Steep learning curve, write your own interfaces
  - https://www.centr.org/docs/2007/05/Tech16_9_Dickinson.pdf
  - http://www.nlnetlabs.nl/publications/hsm/index.html

Example implementation

- Based on Net::DNS::SEC frontend to the BIND dnssec tools

Private Key Maintenance Basic Architecture



Zone DB

Signer client

DNS server

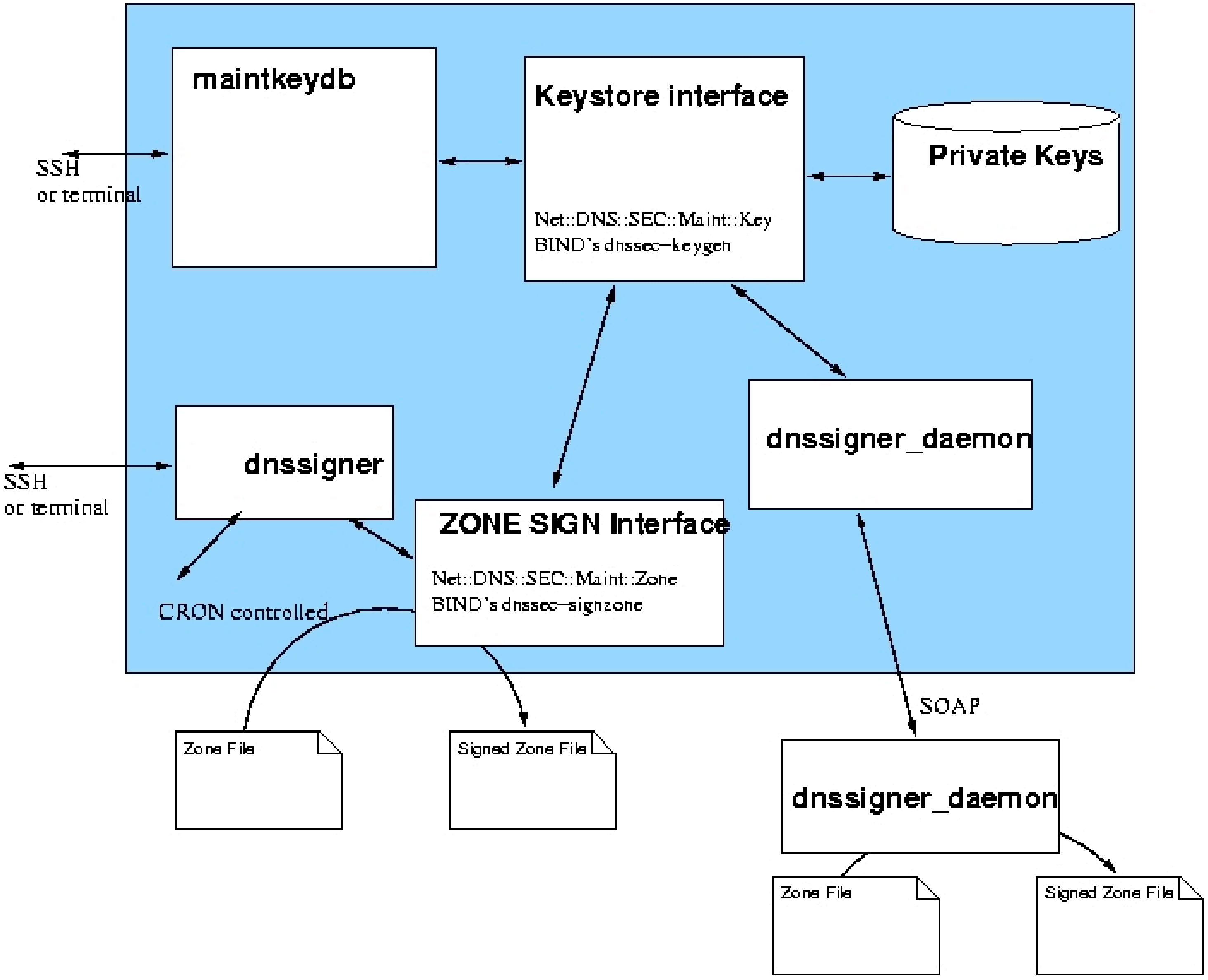
Key maintainer

Key DB and Signer server

KEY Master

Maintaining Keys and Signing Zones

- The KeyDB maintains the private keys
  - It ‘knows’ rollover scenarios
  - UI that can create, delete, roll keys without access to the key material
  - Physically secured
- The signer ties the Key DB to a zone
  - Inserts the appropriate DNSKEYs
  - Signs the the zone with appropriate keys
- Strong authentication


maintkeydb
Keystore interface
Net::DNS::SEC::Maint::Key
BIND's dnssec-keygen
Private Keys
SSH
or terminal
SSH
or terminal
dnssigner
dnssigner_daemon
ZONE SIGN Interface
Net::DNS::SEC::Maint::Zone
BIND's dnssec-signzone
CRON controlled
SOAP
Zone File
Signed Zone File
dnssigner_daemon
Zone File
Signed Zone File

Private Key Maintenance The software

- Perl front-end to the BIND dnssec-signzone and dnssec-keygen tools
- Key pairs are kept on disc in the “BIND format”
- Attribute files containing human readable information
  - One can always bail out and sign by hand.

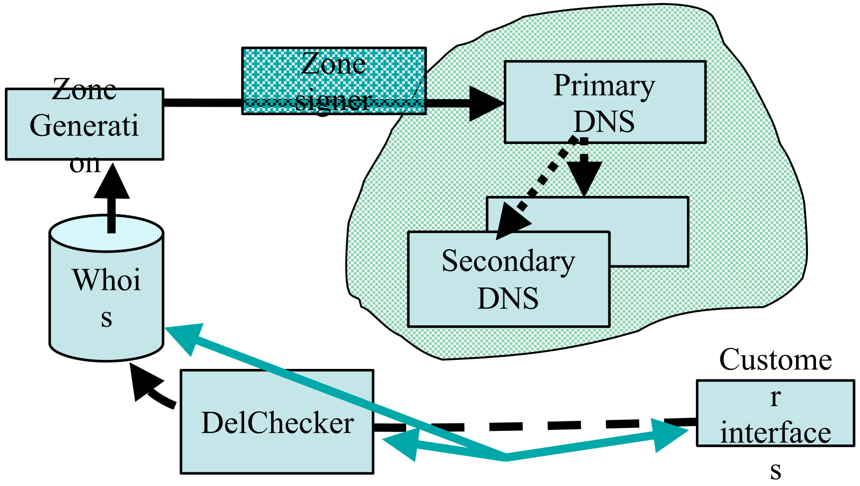
- Works in the RIPE NCC environment, is a little rough edged but available via the www.ripe.net/disi

Example session

```
$ maintkeydb create KSK RSASHA1 2048 example.net
      Created 1 key for example.net
$ maintkeydb create ZSK RSASHA1 1024 example.net
      Created 2 keys for example.net
$ dnssigner example.net
      Output written to :example.net.signed
```

```
$ maintkeydb rollover zsk-stage1 RSASHA1 example.net
```

Presentation roadmap



DNSSEC aware provisioning

- Overview of problem space
  - Architectural changes to allow for DNSSEC deployment
- Deployment tasks
  - Key maintenance
  - DNS server infrastructure
  - Providing secure delegations

Infrastructure

- One needs primary and secondary servers to be DNSSEC protocol aware

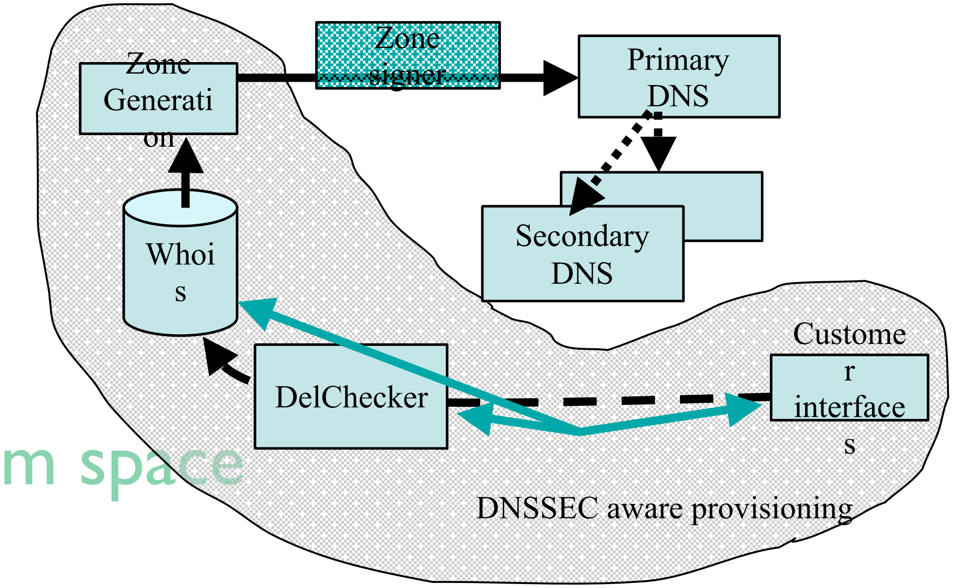
- We had a number of concerns about memory CPU and network load
  - Research done and published as RIPE 352

Conclusion from RIPE 352

- CPU, Memory and Bandwidth usage increase are not prohibitive for deployment of DNSSEC on k.root-servers.net and ns-pri.ripe.net

- Bandwidth increase is caused by many factors
  - Hard to predict but fraction of DO bits in the queries is an important factor
- CPU impact is small, Memory impact can be calculated
- Don’t add DNSKEY RR set in additional

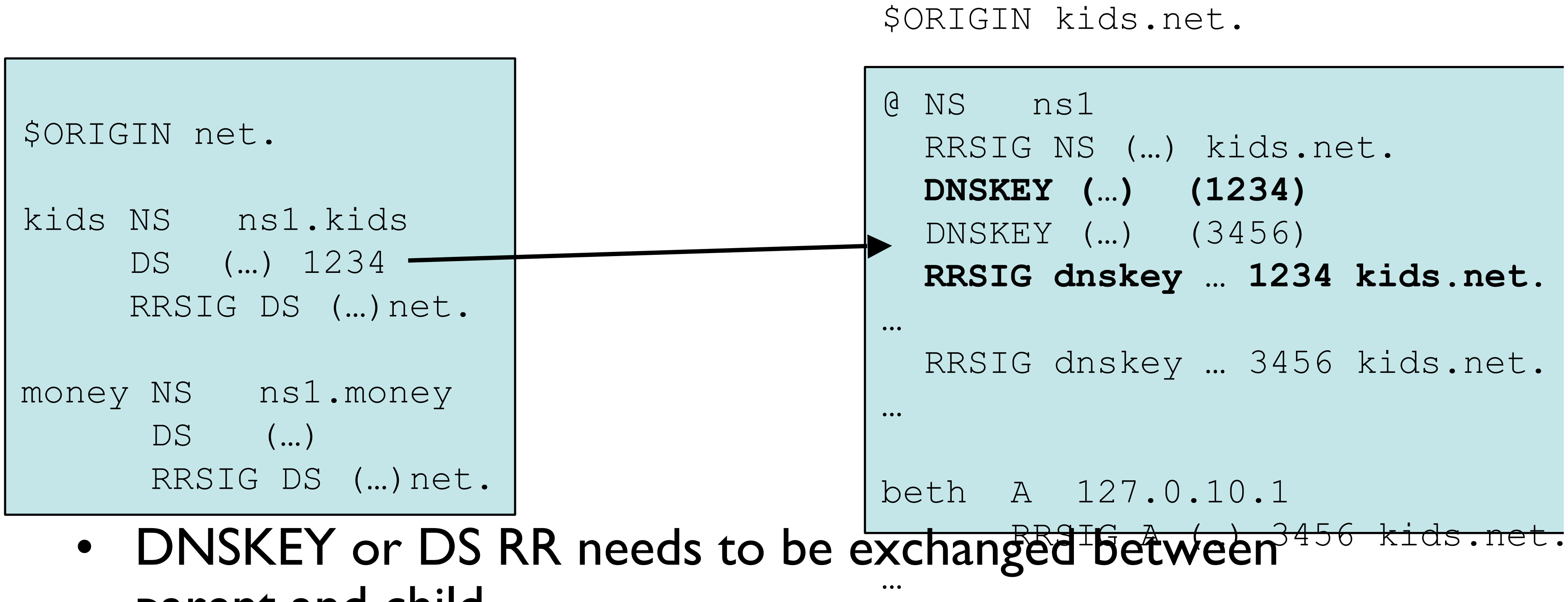
Presentation roadmap



- Overview of problem space
  - DNSSEC in 3 slides
  - Architectural changes to allow for DNSSEC deployment
- Deployment tasks
  - Key maintenance
  - DNS server infrastructure
  - Providing secure delegations

Parent-Child Key Exchange

- In the DNS the parent signs the “Delegations Signer” RR
  - A pointer to the next key in the chain of trust



```
$ORIGIN net.

kids NS    ns1.kids
     DS   (…) 1234
     RRSIG DS (…)net.

money NS    ns1.money
      DS    (…)
      RRSIG DS (…)net.
```

```
$ORIGIN kids.net.

@ NS    ns1
  RRSIG NS (…) kids.net.
  DNSKEY (…)  (1234)
  DNSKEY (…)  (3456)
  RRSIG dnskey … 1234 kids.net.
…
  RRSIG dnskey … 3456 kids.net.
…

beth  A  127.0.10.1
      RRSIG A (…) 3456 kids.net.
…
```

- DNSKEY or DS RR needs to be exchanged between parent and child

Underlying Ideas

- The DS exchange is the same process as the NS exchange
  - Same authentication/authorization model
  - Same vulnerabilities
  - More sensitive to mistakes
- Integrate the key exchange into existing interfaces
  - Customers are used to those
- Include checks on configuration errors
  - DNSSEC is picky
- Provide tools
  - To prevent errors and guide customers

Questions and Discussion