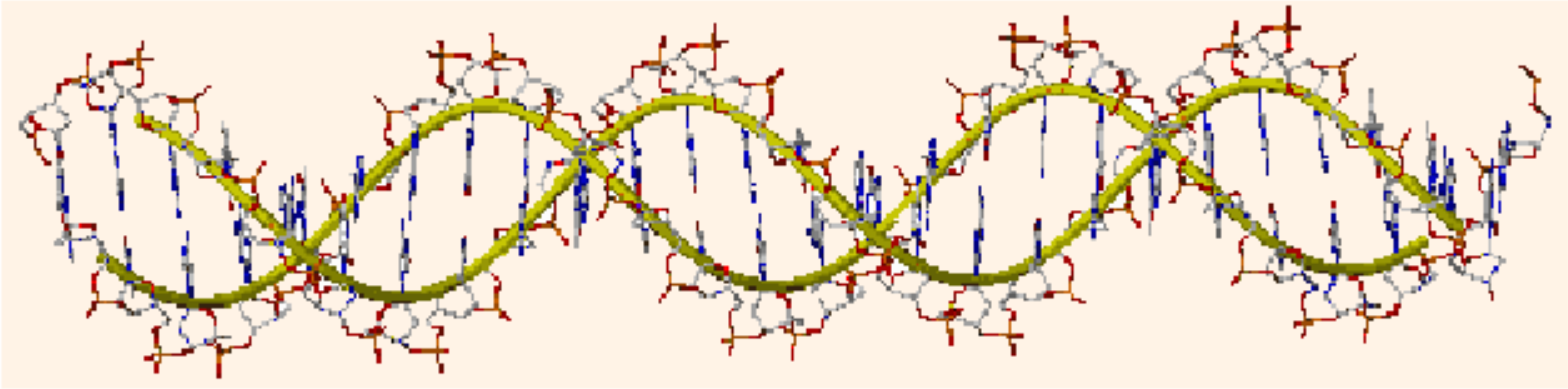
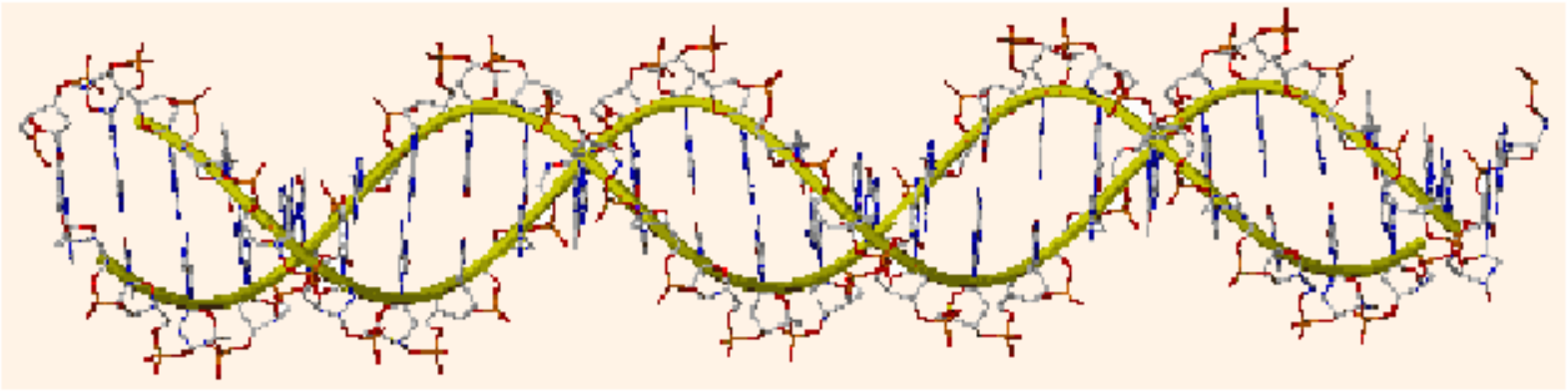
D N S

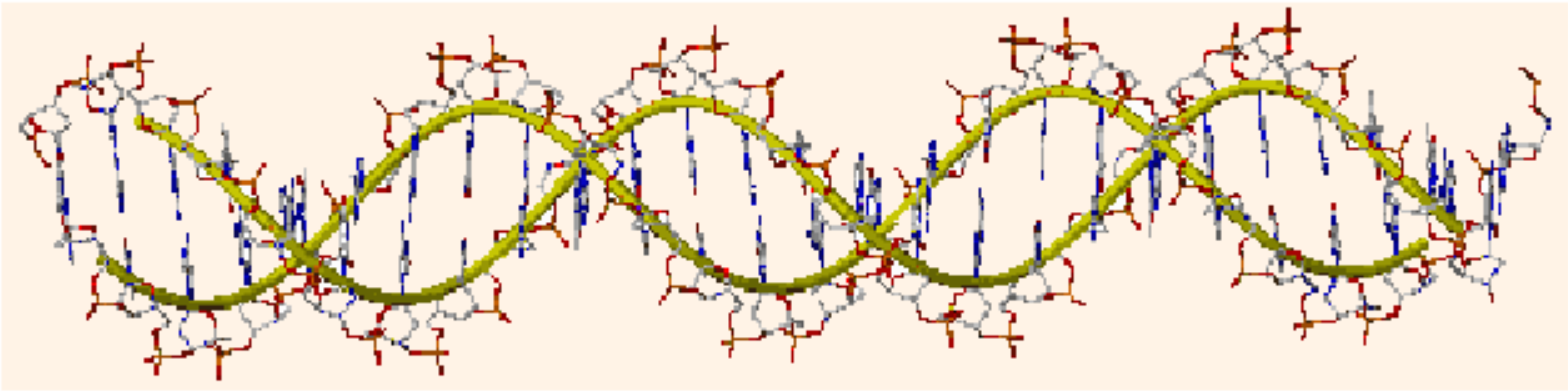


D N S



Desoxyribonukleinsäure?

D N S



Desoxyribonukleinsäure?

Domain Name Service!

Was ist DNS?

- „Telefonbuch“ des Internet.
  Welche IP-Adresse hat der Rechner `host.example.de`?
- Ersatz für die Datei `hosts.txt` aus der Anfangszeit des Internet.
- Verteilte Datenhaltung, hierarchisch strukturiert.
- Basiert auf Domainnamen:
  ```
  host.subdomain.domain.tld
  horst.mobile.example.de
  ```
- Primäre Anwendung:
  - Name zu IP-Adressauflösung (Forward Lookup).
  - IP zu Namensauflösung (Reverse Lookup).
  - Mailrouting (MX).

Anwendungsgebiete

Die Speicherung der Daten erfolgt in anwendungsspezifischen Datensätzen (Resource Records).

- Namensauflösung IPv4 und IPv6

```
www.example.de.   A      1.2.3.4
                  AAAA   fe80::210:5aff:feac:8b1b
```

  A6 Records experimental (RFC3363)

- Reverse Lookups

```
4.3.2.1.in-addr.arpa.        PTR  www.example.de.
b.1.b.8.c....e.f.ip6.arpa.   PTR  www.example.de.
```

  IPv6 reverse delegation (RFC3152)

- Mailrouting (Mail to: `horst@example.de`)

```
example.de   MX 10  mail.example.de.
             MX 20  mail2.example.de.
```

Anwendungsgebiete (2)

- Service Location Records (RFC2052)

```
_ldap._tcp.example.de   SRV  10 1  389 ldap.example.de.
```

  Noch nicht sehr weit verbreitet (Microsoft, Apple Rendezvous).

- Location Records

```
horst.example.de   LOC  40 2 0.373 N 105 17 23.231 W 140m
```

- E.164 Telefonnummer zu URI-Mapping, ENUM (RFC2916).
  Telefonnummer: +49 (0)6633/5739

```
9.3.7.5.3.3.6.6.9.4.e164.arpa.  \
    NAPTR 100 10 "u" "sip+E2U"  "!^.*!sip:info@ex.de!i"    .
    NAPTR 102 10 "u" "smtp+E2U" "!^.*!mailto:info@ex.de!i" .
```

- NAPTR (RFC3404) Siehe oben.

- Public-Keys (KEY) und Zertifikate (CERT)

Anwendungsgebiete (3)

- SSH-Fingerprints (draft-ietf-secsh-dns-04.txt)

```
host.example.de   SSHFP 2 1  123456789abcdef67890....
```

- Adress Prefix Listen (APL) (RFC3123)

```
net.example.de   APL 1:145.253.0.0/16 1:145.254.0.0/16
```

- Key Exchange Record (KX) (RFC2230)

```
example.de   KX 10  kdc.example.de.
             KX 20  kdc.example.de.
```

- Secure DNS (RFC2535)
  `DNSKEY`, `RRSIG` und `NSEC`

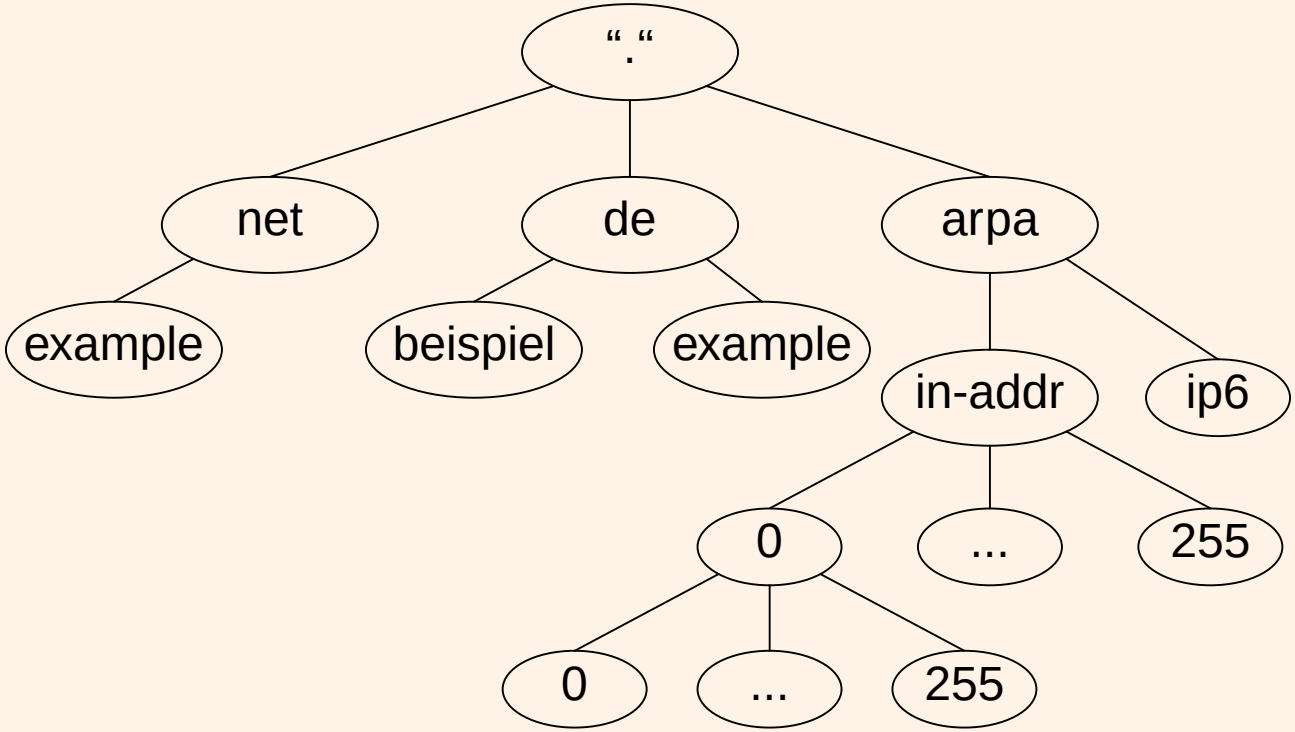
Struktur des Domain Name System

- Für jede Domain existieren sog. authorisierte Nameserver[1].
  Diese stellen alle Informationen über eine Domain zur Verfügung.
- Auf untergeordnete Domains wird über NS-Records verwiesen.

```
subdomain.example.de.   NS  ns1.example.de.
```

- Von der übergeordneten Domain (tld) existieren Verweise auf diesen Nameserver (Ebenfalls über NS-Records).
- An der Spitze dieser Hierarchie stehen die 13 Root-Nameserver.
  `A.ROOT-SERVERS.NET` – `M.ROOT−SERVERS.NET` (http://www.root-servers.org)
- Nur zwei der Root-Server stehen in Europa (GB, S), einer in Asien.
  Tatsächliche Anzahl durch Anycasting größer (s.u.)
- Einzelne Server (z.B. `F.ROOT-SERVER.NET`) sind weltweit gespiegelt
  Zugriff über „BGP-Anycast“ (Hierachical Anycast).

Hierarchischer Aufbau



- Pro Domain mindestens zwei authorisierte Nameserver.
- Aber: Mehrere Domains können auf einem Nameserver gehostet werden.

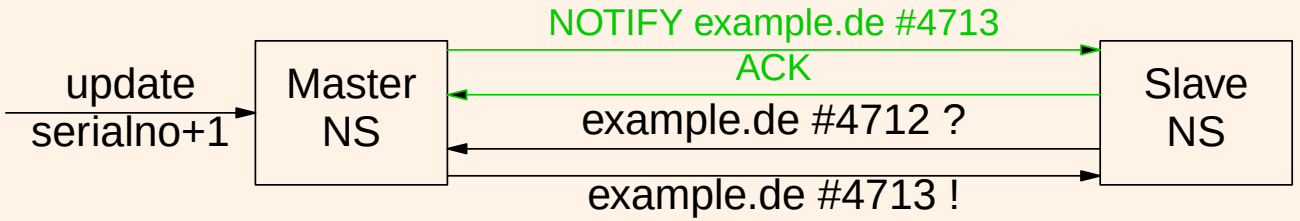
Redundanzkonzept

- Pro Domain mehrere authorisierte Nameserver.
- Einer ist Master (Primary), alle anderen sind Slave Server (Secondary).
- Pflege der Domaindaten ausschließlich auf dem Master (Über die Zonendatei oder über „dynamic update“).
- Die Slave Server holen die Domaindaten über einen Zonentransfer (AXFR).
- Die Verteilung erfolgt in festen Intervallen (Refresh) oder durch den Master getriggert (NOTIFY).
- Neuere Implementierungen unterstützen inkrementelle Zonentransfers (IXFR).
- Authentisierung über kryptographische Verfahren möglich (TSIG)

Redundanzkonzept (Zonentransfer)

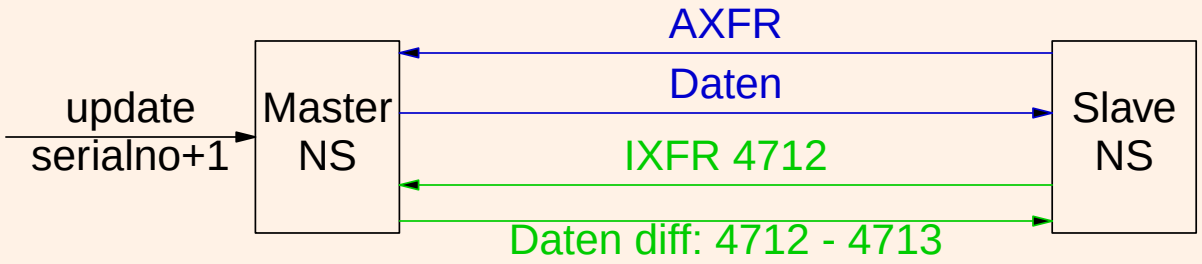
Signalisierung:

Abgleich der Seriennummer: Zeitbasiert oder durch NOTIFY getriggert.



Datentransfer:

Kompletter (AXFR) oder inkrementeller (IXFR) Zonentransfer.



Namensanfragen

- Anfragen an Nameserver werden von vielen Programmen benötigt (telnet, ftp, Internet-Browser, MTA, usw.).
- Die Clientfunktionalität der Namensanfrage wurde in Form einer Bibliotheksroutine implementiert (sog. Resolver Library).

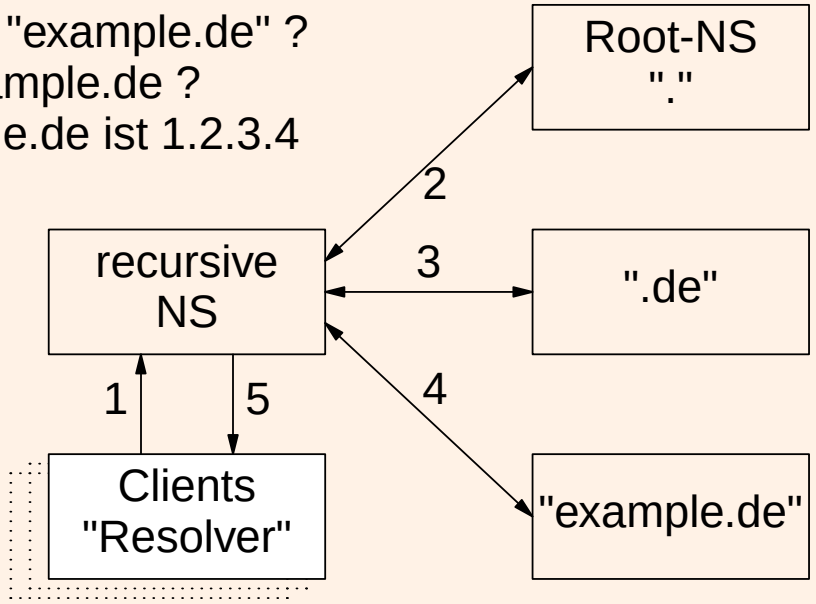
```
# include <resolv.h>
res_query (const char *dname, int class, int type, uchar *answer, int alen);
```

- Die IP-Adressen(!) der zu verwendenden Nameserver sind in der Datei `/etc/resolv.conf` hinterlegt.
- Ein Clientsystem stellt eine Anfrage an den ersten aufgeführten NS und wartet 5 Sekunden auf eine Antwort.
- Der Nameserver muß für den Client die Antwort ermitteln.
- Bekommt der Client keine Antwort verwendet er den nächsten NS in seiner Liste.

Resolving Nameserver

- Nameserver, die im Auftrag von Clients Namen auflösen, werden „resolving“ oder auch „rekursive“ Nameserver genannt.
- Der Resolving-Nameserver muß die IP-Adressen der Root-Server kennen!

1) Welche IP hat www.example.de ?
2) Wer ist authoritative für ".de" ?
3) Wer ist authoritative für "example.de" ?
4) Welche IP hat www.example.de ?
5) Die IP von www.example.de ist 1.2.3.4



Resolving Nameserver (2)

- Rekursive Nameserver speichern die Ergebnisse in einem Cache
  (Caching-only Nameserver)
- Die Dauer der Speicherung wird durch den TTL-Wert (Time to Live) bestimmt.
- TTL- Wert wird durch den authoritative NS vorgegeben
  Er sollte von dem Caching-Nameserver nicht vergrößert werden
- Neuere Implementierungen unterstützen auch negatives Caching.
- Immer mehr Domains verwenden extrem kurze TTLs:
  Akamai, DynDNS.org, usw.
- ... bis hin zur faktischen Abschaltung des Caches :-(

```
www.bahn-net.de.        0     IN A   193.25.241.243
```

DNS Protokoll

- Client − Server Protokoll
- Transport Protokoll ist sowohl UDP als auch TCP!!!
- Zonentransfers: TCP (Destination Port 53)
- Update Requests: UDP oder TCP (Destination Port 53)
- Query: UDP oder TCP (Destination Port 53)
- Paketgrösse bei UDP: max. 512 Byte
- Durch EDNS0 (RFC2671) kann die max. UDP-Paketgröße ausgehandelt werden (Bis 4096 Byte)
- Für die Verwendung von IPv6-Adressen und Secure-DNS notwendig

Server Implementierungen

- ISC − Berkley Internet Name Daemon (BIND)
  - 4.9 Unsicher, veraltetes Konfigfile
  - 8.4.5 Schnell, keine Weiterentwicklung, nur Bugfixes
  - 9.2.4 Saubere Implementierung, viele Features (views, IPv6), vergleichsweise langsam.
  - 9.3.0 Vollständige Secure-DNS Implementierung
- djbdns („Sicherer“ Nameserver von D.J.Bernstein)
- nsd (Authoritative only NS von NLnet LABS; Secure-DNS Unterstützung)
- Microsoft (Properitär bei dyn. Update und Zonentransfers)
- Viele andere...

Server Implementierungen (2)

Anteil der Server Software bei den TLD Nameservern[2].

- ~ **56 %** BIND 4.9.3+ und BIND 8
- ~ **34 %** BIND 9
- ~ **4,5 %** NN
- ~ **2,7 %** UltraDNS
- ~ **0,9 %** BIND 4
- ~ **0,3 %** djbdns

---

2. Brad Knowles „Domain Name Server Comparison“ Januar 2003 Ripe44 (*http://www.ripe.net/ripe/meetings/archive/iripe-44/presentations/ripe44-dns-dnscomp.pdf*)

Client Implementierungen

- Nur in Form der Resolver-Bibliothek.
- Retry-Verhalten im Fehlerfalle unterschiedlich.

  Aber: Bei negativer Antwort wird **kein** zweiter NS befragt.
  Alle NS sind gleichwertig!
- Unix
  Maximal 3 Nameserver, maximal 3 Wiederholungen:

Vers.	Wiederh.	Anzahl Nameservereinträge 1	2	3
> 8.2	1	(1x) 5s	(2x) 5s	(3x) 5s
		(1x) 10s	(2x) 5s	(3x) 3s
	Total	15s	20s	24s
< 8.2	2	(1x) 20s	(2x) 10s	(3x) 6s
	3	(1x) 40s	(2x) 20s	(3x) 13s
	Total	75s	80s	81s

Client Implementierungen (2)

- Win95/Win98
  ?

- Win NT4.0

  Caching des Ergebnisses (max. TTL) pro anfragendem Prozeß.

  Ab Servicepack 4:

  — Anfrage an den ersten NS mit einem Timeout von 1 Sekunde.
  — Danach, Anfrage an alle(!) Nameserver (Timeout 2 Sekunden).
  — Bis zu vier Wiederholungen mit Verdopplung der Wartezeit.

- Windows 2000

  Wie NT4.0 SP4

  Zusätzlich: Registrierung des Rechnernamens über Dynamic Updates.

```
04-Nov-2004 20:09:01.389 update: info: client 217.252.122.205#1108: \
       updating zone 'hznet.de/IN': update unsuccessful: w2kws1.HZNET.DE/A: \
       'RRset exists (value dependent)' prerequisite not satisfied (NXRRSET)
```

Tools

Diverse Hilfsprogramme zum Testen:

`host` Anwenderfreundlich

```
$  host www.cisco.com
www.cisco.com has address 198.133.219.25
```

`nslookup`

Bei Tests manchmal problematisch.

```
$  nslookup www.cisco.com
Server:    10.14.33.67
Address:   10.14.33.67#53

Non-authoritative answer:
Name:  www.cisco.com
Address: 198.133.219.25
```

`dig` Bestes Tool zum Debuggen

Allerdings Raw„-Output“:

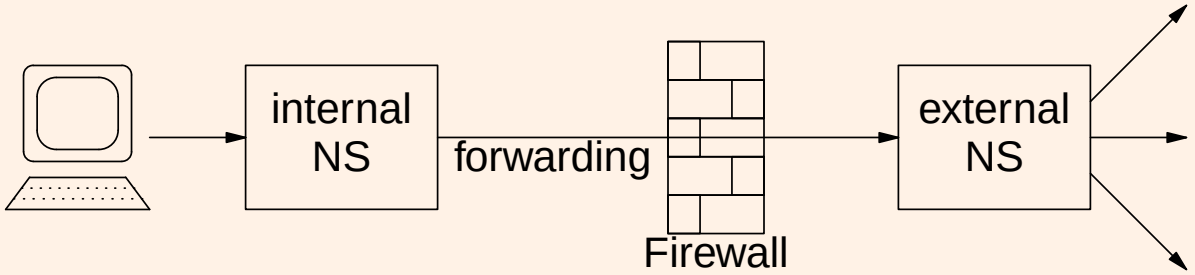
Tools (2)

```
$  dig www.cisco.com
; <<>> DiG 9.3.0s20021217 <<>> www.cisco.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3826
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.cisco.com.             IN   A

;; ANSWER SECTION:
www.cisco.com.          72996    IN   A    198.133.219.25

;; AUTHORITY SECTION:
cisco.com.          61588    IN   NS   ns1.cisco.com.
cisco.com.          61588    IN   NS   ns2.cisco.com.

;; ADDITIONAL SECTION:
ns1.cisco.com.          35547    IN   A    128.107.241.185
ns2.cisco.com.          72996    IN   A    192.135.250.69

;; Query time: 4 msec
;; SERVER: 10.14.33.67#53(10.14.33.67)
;; WHEN: Thu Jan 23 15:54:33 2003
;; MSG SIZE  rcvd: 115
```

Feature: Forwarding Nameserver

- Sind „nicht rekursive“ Nameserver (Proxy NS).
- Alle Anfragen werden an einen anderen Nameserver weitergeleitet
  (seit Bind9.2 auch domainbasiert)

Beispiel: Firewall



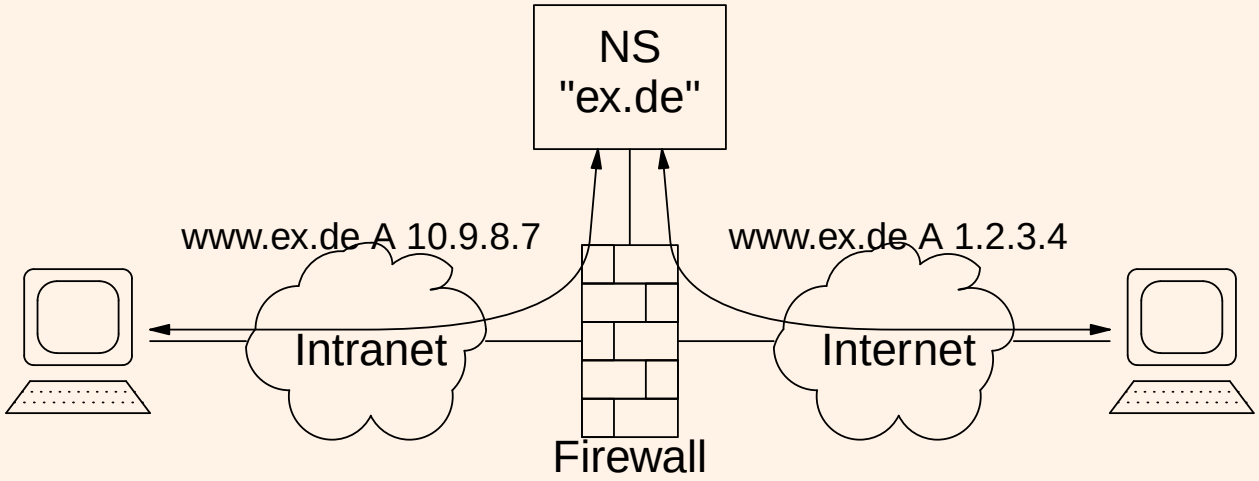
Firewall Regel (Stateless packetfilter)

	Source		Destination	
	IP	Port	IP	Port
query	intNS	any (>1024)	extNS	53 TCP/UDP
answer	extNS	53 TCP/UDP	intNS	any (>1024)

Feature: Views

- Views unterstützen unterschiedliche „Sichten“ auf Domains.
- „Interne“ Clients bekommen andere Ergebnisse als „externe“.

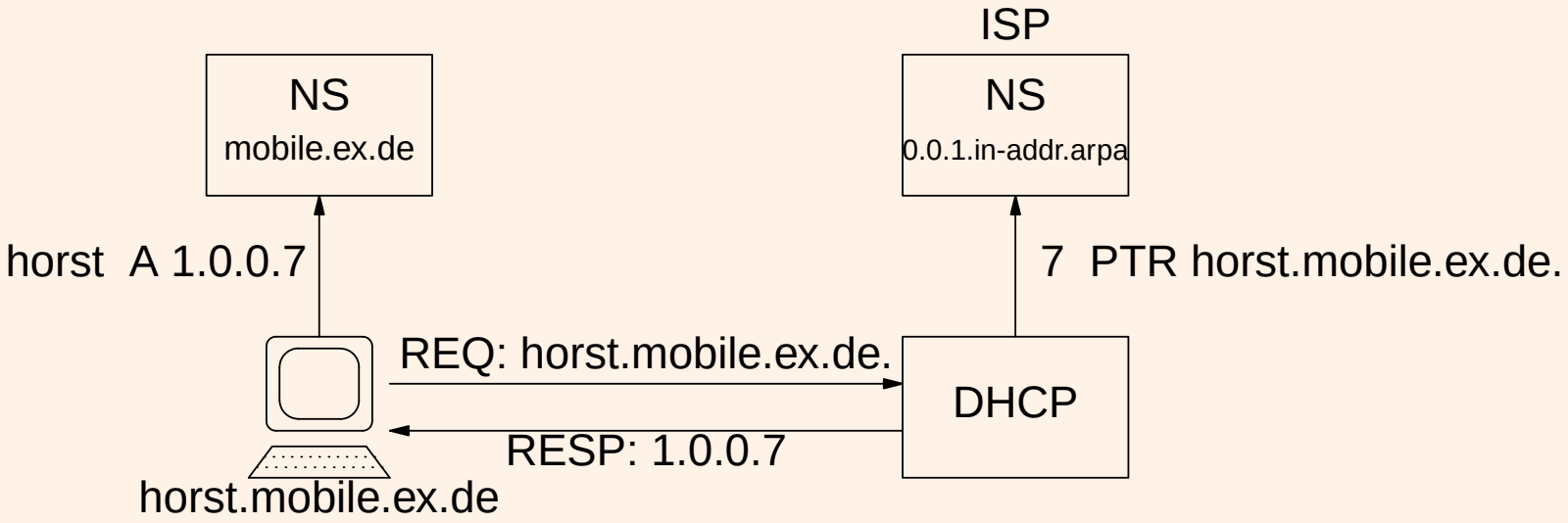
Beispiel: Firewall



Feature: Dynamic Update

- Erlaubt die dynamische Modifikation von Domaininhalten durch authentisierte Clients.
- Authentisierung entweder über TSIG (Shared MD5-Keys) oder SIG(0) (Public-Key). Microsoft verwendet GSS-API (Kerberos).

Beispiel: Laptop



Feature: Secure DNS (TSIG)

Transaktions Signaturen (TSIG)

- Shared Secret (HMAC-MD5).
- Sichert Zonentransfers zwischen Master und Slave.
- Sichert dynamische Updates zwischen z.B. DHCP-Server und Master Nameserver.
- Keine Verschlüsselung!
  Öffentliche Daten!
- Authentisierung und Integritäts-Check.
  Wer bin ich? Sind die Daten modifiziert worden?
- Voraussetzung: Synchronisierte Uhren (diff max. 5 Min).

Feature: Secure DNS (RRSIG)

Signatur der Domaineinträge

- Sicherstellung der Authentizität der Nameserver Antworten
- Parent signiert Domain Schlüssel
- Domain Schlüssel signiert Domaineinträge
- Chain of Trust ausgehend von der Root-Zone
  Nur Public-Key der Root-Zone zur Verifizierung notwendig
- Weltweit noch schwierig zu etablieren
  DENIC wird es nicht einführen solange Zonewalk möglich ist
- Im Enterprise Umfeld aktuell einsetzbar
  BIND 9.3.0, NSD 2.0
- Alternativen zu hierarchischem Model in Entwicklung (DLV)

References

Paul Albitz, Cricket Liu
„DNS and BIND“, O’Reilly & Associates, 1998

Brad Knowles
Domain Name Server Comparison
(*http://www.shub-internet.org/brad/papers/dnscomparison/*)

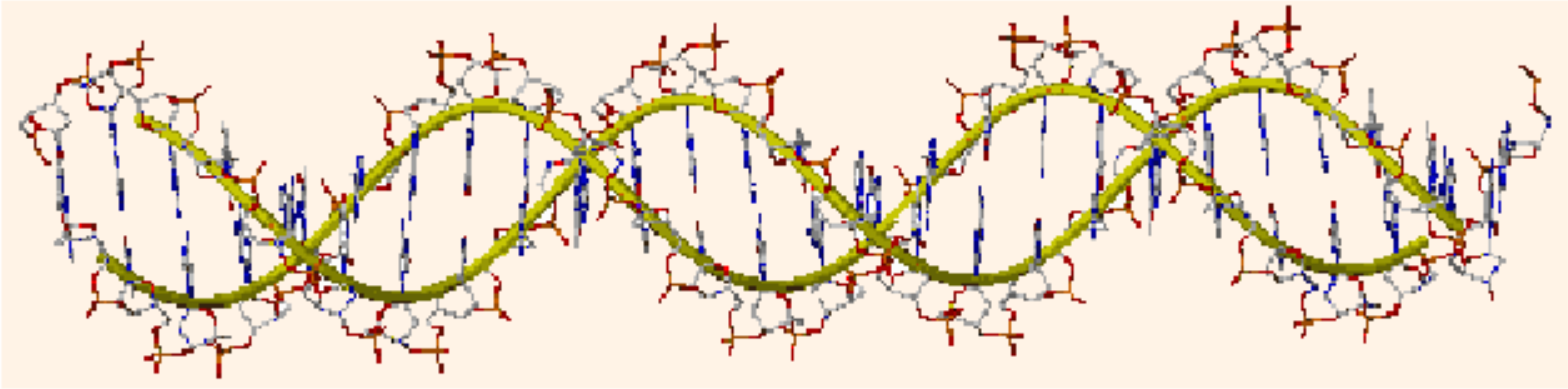
Nominum
BIND v9 Administrator Reference Manual
(*http://www.nominum.org/content/documents/bind9arm.pdf*)

Olaf Kolkman, RIPE DISI
DNSSEC Howto Version 1.3
(*http://www.ripe.net/disi/Course/TechCourse-1.3-release.pdf*)

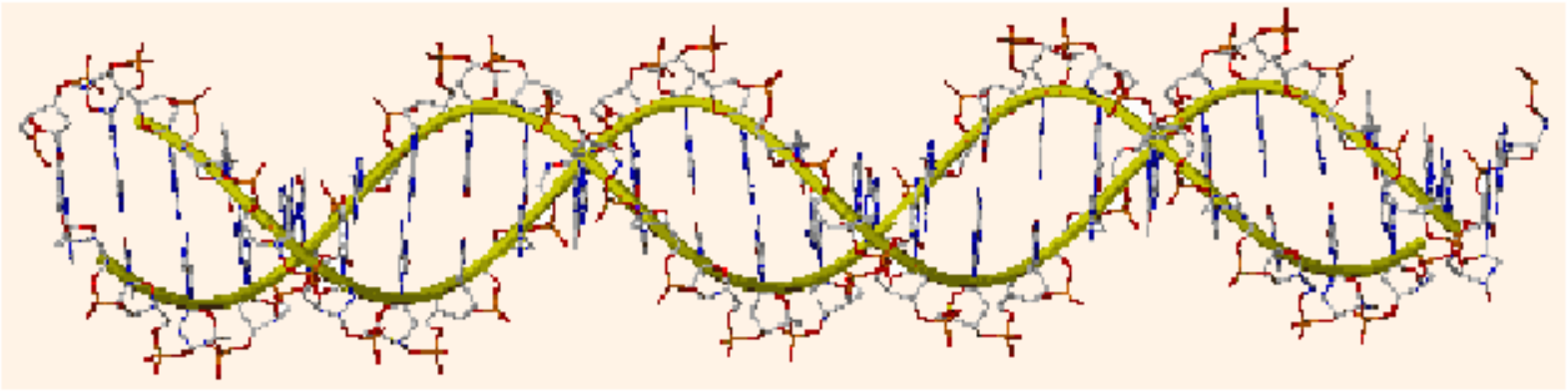
Carsten Schiefner, RIPE 44
ENUM − A status update
(*http://www.ripe.net/ripe/meetings/archive/ripe-44/presentations/ripe-44-dnr-enum/*)

RFC
1034, 1035, 2052, 2230, 2535, 2671, 3152, 3363, 3404, ...

Fragen ?

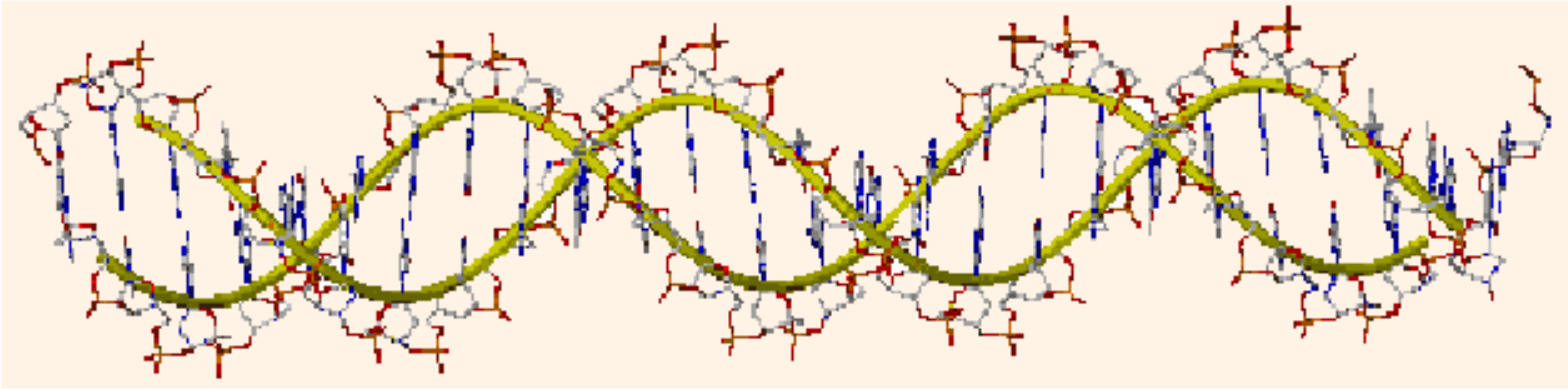


Fragen ?



http://www.hznet.de/dns/einfdns.pdf

Fragen ?



http://www.hznet.de/dns/einfdns.pdf

Danke!

CONTENTS