```
DNS
===

Elemente von DNS:

1) Namensraum und Resource Records
----------------------------------

- baum-strukturierter Namensraum:
  - Wurzel "." ("oben")
  - Aufzählung für absolute Namen fängt beim "untersten" Element an
  - Trennung der einzelnen Pfadbestandteile mit Punkten
  - Kennzeichnung absoluter Namen: Name endet mit "."
  - Name max. 63 Zeichen, gesamter Name max. 255 Zeichen, Groß-/Kleinschr. egal
  - TLDs (s. Link auf Webseite):
    - ccTLDs (knapp 250 Stück)
    - generic (com, net, org, edu, gov, mil und 7 "neue" TLDs)
    - administrative (arpa)

- jedem Element _können_ Informationen "angeheftet" werden: Resource Records

- wichtige Resource Record (RR) Types sind:
  SOA   identifiziert den Start einer authoritativen Zone (start of authority)
  NS    authoritative Nameserver dieser Domain
  A     IP-Adresse, die diesem Namen zugeordnet ist
  MX    identifiziert Mail eXchanger dieser Domain (jeweils Name und Priorität)
  PTR   ein Pointer auf einen anderen Teil des Namensraums
  CNAME identifiziert den "canonical name" zu einem Alias ("exklusiv"-RR)

- reverse Lookup über in-addr.arpa-Domain (PTR-Records) (Beispiel: IP: 1.2.3.4,
  Host: demo.de.; Pointer-RR: 4.3.2.1.in-addr.arpa. --PTR--> demo.de.)

- beliebter Denkfehler:
  Assoziation mit Netzen, tu-berlin.de entspräche 130.149.0.0/16, eine solche
  Zuordnung ist nicht zwangsläufig notwendig, kann aber existieren

2) Name Server
--------------

- Serverprogramme, die DNS-Informationen liefern (i.d.R. Frage => Antwort)

- Server hält alle Daten für einen oder mehrere Teilbäume vor, diese Teilbäume
  heißen Zonen, Server ist authoritativer Server bezüglich seiner Zonen

- "Cuts" trennen im Namensraum die einzelnen Zonen voneinander

- es gibt mind. zwei authoritative Nameserver pro Zone (Ausfallsicherheit):
  Master <-> Slave-Server, Abgleich über sog. Zonentransfer (AXFR)

- Delegation der Zuständigkeit für "untergeordnete" Zonen über NS-RR
   (Bsp.: DeNIC hat 3 Einträge: tu-berlin.de. --NS--> je_ein_TU-Nameserver)

- Root-Nameserver sind weltweit in jeden Nameserver "eingebaut": dig . ns

- Normal-Modus: Client muß selbst den zuständigen Nameserver finden
  Rekursiv-Modus: Befragter Nameserver fragt sich statt des Clients durch
  Rekursion ist optional, C+S müssen sich auf Rekursion einigen

- Konzept des Caching (only) Nameserver

- DNS-Forwarding: Untergeordnete Nameserver befragen für andere Domains _immer_
  die gleichen übergeordneten Server (muß im Nameserver konfiguriert werden)
```



```
3) Resolver
-----------

- Programmcode zum Befragen von Nameservern

- i.d.R. Bestandteil des Betriebssystems
  bei Linux in der glibc enthalten, Konfiguration über /etc/resolv.conf
  (standardmäßig zu befragende Nameserver, Domain-Suchreihenfolge)


Beispiele
---------

Nameserver befragen: dig <domain.name> [SOA|NS|A|MX|PTR|CNAME]
Reverse Lookup:      dig -x <IP.Adr.es.se>

hodgson.pi.tu-berlin.de.:

.                - SOA: a.root-servers.net. nstld.verisign-grs.com.
|                - NS: 13 Root-Nameserver
|
de               - SOA: dns.denic.de. ops.denic.de.
|                - NS:  11 Nameserver
|
tu-berlin        - SOA: names.zrz.tu-berlin.de. noc.zrz.tu-berlin.de.
|                - NS:  3 Nameserver
|                - MX:  3 Mailserver
|                - PTR: 0.0.149.130.in-addr.arpa.
|
pi               - SOA: names.zrz.tu-berlin.de. noc.zrz.tu-berlin.de.
|                - NS:  3 Nameserver
|                - MX:  3 Mailserver
|
hodgson          - A:   130.149.47.18
                 - MX:  3 Mailserver

www.cfd.tu-berlin.de.:

.                - SOA: a.root-servers.net. nstld.verisign-grs.com.
|                - NS: 13 Root-Nameserver
|
de               - SOA: dns.denic.de. ops.denic.de.
|                - NS:  11 Nameserver
|
tu-berlin        - SOA: names.zrz.tu-berlin.de. noc.zrz.tu-berlin.de.
|                - NS:  3 Nameserver
|                - MX:  3 Mailserver
|                - PTR: 0.0.149.130.in-addr.arpa.
|
cfd              - (null)
|
www              - CNAME: hodgson.pi.tu-berlin.de.


whois
-----

Datenbank mit Kontaktinformationen zu Domains und IP-Adreßbereichen

Beispiele:       - whois tu-berlin.de
                 - whois 130.149.47.15
```

Mail

E-Mail-Adresse

- Aufbau: <local-part>@<mail-domain>

- <mail-domain> muß einen A- oder MX-Resource Record haben

- Gültigkeit des <local-part> kann i.d.R. nur der Empfangs-MTA überprüfen

Aufbau einer E-Mail

- Envelope (entspricht Brief-Umschlag)
  - wird nur während des Transports benutzt, ist später _nicht_mehr_zu_sehen_
  - enthält eine (meist ungeprüfte) Absenderadresse, einige Mailserver prüfen, ob die Domain einen A- oder MX-RR hat (z.B. ZRZ), falls nein, wird die Mail nicht weitergeleitet!
  - enthält eine oder mehrere Empfängeradressen
    => NUR DIESE SIND FÜR DIE ZUSTELLUNG RELEVANT!

- Header (entspricht Brief-Kopf)
  - _kann_ _optional_ _beliebige_ Absender- oder Empfängeradressen beinhalten, diese sind für den Mailtransport _irrelevant_
  - optional Zeitstempel und Betreff möglich
  - i.d.R. Message-ID zur eindeutigen Identifizierung einer Mail
  - oftmals Return-Path: <Envelope-Sender> und Delivered-To: <Envelope-Empf.>
  - von jedem MTA: Received-Zeilen: Protokollierung des Mail-Transports

- Body (entspricht Brief-Inhalt)
  - eigentliche Mail, optional mit einem oder mehreren Anhängen
  - vom Header mit einer Leerzeile getrennt

=> Mailprogramm zeigt maximal Header und Body an, nicht den Envelope

=> Transportprotokoll (SMTP): ASCII, d.h. Binärdaten müssen umkodiert werden, Beschränkung der Mailgröße beachten (meist wenige MB)

Der Weg einer Mail

1) Mail mit Mailprogramm (MUA: Mail User Agent) schreiben

2) MUA übergibt die Mail zur weiteren Zustellung an einen Mailserver (MTA: Mail Transfer Agent)

   Zwei Möglichkeiten dafür:
   1) lokal ("/usr/sbin/sendmail", nicht zu verwechseln mit dem MTA sendmail)
      Auf dem gleichen Rechner, auf dem der MUA läuft, muß der MTA laufen.

   2) Netz (SMTP: Simple Message Transfer Protocol)
      Es wird ein beliebiger MTA übers Netz per SMTP angesprochen. Dieser MTA kann auf dem gleichen Rechner wie der MUA laufen oder auf einem anderen.

   MUAs wie Pine, Elm und Mutt benutzen i.d.R. /usr/sbin/sendmail. Bei MUAs, die direkt einen MTA per SMTP ansprechen, muß dieser vom Benutzer konfiguriert werden (z.B. Netscape/Mozilla).

3) Versand durch Mailserver

   - Ein Mailserver betrachtet eine oder mehrere Domains als "lokal".

     Für diese Domains ist er zuständig, d.h. er leitet Mails i.d.R. nur dann weiter, wenn sie:
     - von einem lokalen Benutzer kommen oder
     - an einen lokalen Benutzer gehen

     Spam:
     MTAs, die Mails von externen Benutzern an externe Benutzer weiterleiten, werden als "open relay" bezeichnet und sind eine potentielle Spam-Quelle. Bekannte open relays werden in schwarzen Listen gesammelt (RBLs: Realtime Blackhole Lists), manche MTAs leiten Mails von open relays nicht weiter! Wähleingänge von Providern werden teilweise auch schon geblockt, deshalb sollte man immer den SMTP-Server seines Providers zum Mailversand nutzen.

   - Ein Mailserver hat einen "Posteingang" (lokaler und/oder SMTP-Server), über den ihm Mails zur weiteren Zustellung eingereicht werden können (s. Punkt 2). Das machen lokale Benutzer, die an externe oder lokale Benutzer schreiben wollen (lokal und/oder SMTP) und externe Benutzer, die an lokale Benutzer schreiben wollen (SMTP).

   - Nur lokale Benutzer eines MTAs sind authorisiert, über ihn Mails an externe Benutzer weiterleiten zu lassen.

     Diese Authorisierung kann erfolgen durch:
     - nichts
       (Server leitet _alle_ Mails weiter, ist damit ein offenes Relay)
     - IP-Adresse
       (oft bei Mailservern von Internet-Providern, z.B. Uni)
     - SMTP-after-POP
       (Mails müssen erst per POP3-Protokoll _abgerufen_ werden)
     - SMTP-Auth
       (optionale Erweiterung des SMTP-Standards um Login/Paßwort)

     Eine Absenderadresse mit der lokalen Domain genügt _nicht_!

   - Verarbeitung im Mailserver:
     - Ziel-Domain des Mail-Empfängers = Lokale Domain des MTAs?

     - Ja (= Mail an lokalen Benutzer):
       - Mail wird im lokalen Postfach dieses lokalen Benutzers abgelegt. Von dort kann der Benutzer sie abholen (s. Punkt 4).

     - Nein (= Mail an externen Benutzer):
       - Ein Mailserver benutzt einen "Postausgang" (SMTP-Client), um Mails für externe Benutzer an deren zuständige "lokale" MTAs weiterzuleiten.
       - MX- oder A-RR der Ziel-Domain liefern die zuständigen Ziel-MTAs. Es kann mehrere MTAs für eine Domain geben, die durch unterschiedliche Prioritäten als Haupt- und Backupserver gekennzeichnet sein können. Zunächst wird versucht, die Mail an einen zuständigen Haupt-MTA auszuliefern, d.h. einen Server mit der höchsten Priorität (=niedrige Zahl). Ist dieser nicht erreichbar oder verweigert die Annahme aufgrund Überlastung, wird eine Zustellung an die Backup-MTAs versucht. Sind auch diese nicht erreichbar, wird meist noch mehrere Tage lang versucht, die Mail doch noch zuzustellen (es gehen dann manchmal entsprechende Warnmails an den Absender).

   => Im einfachsten Fall geht eine Mail folgenden Weg:

   MUA ---(lokal/SMTP)---> Sender-MTA ---(SMTP)---> Empfänger-MTA

```
   => Es kann auch Relay-Server geben, die auf Sender- oder Empfängerseite
      Mails an fest vorgegebene MTAs weiterleiten.

      Beispiel: Mail wird lokal aus dem CIP-Pool von
      demo@henry.pi.tu-berlin.de an oergel@pi.tu-berlin.de geschrieben:

      1. MTA: lokaler MTA auf cip50xx
         -> Weiterleitung an festen Relayserver auf Absenderseite: henry

      2. MTA: henry.pi.tu-berlin.de
         henry ist Relayserver für alle cip50xx-Rechner
         -> Weiterleitung an weiteren, festen Relayserver: ZRZ

      3. MTA: mail[2].zrz.tu-berlin.de ("dig tu-berlin.de mx")
         Zentrale, uni-weite Relayserver
         pi.tu-berlin.de ist lokale Domain für diese MTAs

         (ansonsten würde hier die Ermittlung des Ziel-MTAs per MX-/A-Anfrage
          ans DNS erfolgen und die Mail an einen entsprechenden Ziel-MTA
          weitergeleitet werden = Übergang von Sender- zu Empfängerseite)

         -> Umschreiben des "PI-Alias" oergel@pi.tu-berlin.de zu
            oergel@prandtl.pi.tu-berlin.de (nur Envelope-Adresse)
         -> Weiterleitung an prandtl.pi.tu-berlin.de

      4. MTA: prandtl.pi.tu-berlin.de
         Ziel-MTA erreicht, Zustellung ins lokale Postfach des Users oergel

   => Mail wird solange weitergeleitet, bis sie an einem MTA ankommt, der den
      Empfänger als lokal betrachtet und die Mail lokal zustellt oder bis ein
      Mailserver die Mail als unzustellbar zurückweist ("bounced").

4) Abholung

   - Im Postfach eines MTAs kann eine Mail lokal (in /var/[spool]/mail/<uid>)
     gelesen werden oder über einen zusätzlichen POP3-Server von einem POP3-
     Client des Benutzers abgerufen werden:
     - POP3 ist wie SMTP ein ASCII-basiertes Protokoll
     - Mail kann auf Server verbleiben oder dort gelöscht werden
     - Default: Klartextübermittlung von Login/Paßwort und Mail-Inhalt,
       Absicherung z.B. mit SSL-Verschlüsselung möglich

Mail-Rückverfolgung
-------------------

Jeder MTA fügt einer transportieren Mail einen "Poststempel" hinzu. Dazu wird
eine "Received:"-Zeile in den Header eingefügt. Neue Received-Zeilen stehen da-
bei weiter oben. Daher steht der erste MTA ganz "unten", der letzte, d.h. der
jeweils lokale MTA des Empfängers, ganz "oben".

Meistens enthalten Received-Zeilen die folgenden Angaben:
- Empfangender MTA (der jeweils gerade aktuelle)
- Sendender MTA (IP-Adresse und DNS-Name, falls abweichend: HELO-Name, s. SMTP)
- Envelope-Empfänger
- Queue-ID der Mail in diesem MTA
- Zeitstempel mit Angabe der Zeitzone
- MTA-Software

Spammer fälschen gerne Header-Zeilen, um den Weg einer Mail zu verschleiern.
Dabei werden auch falsche Received-Zeilen eingesetzt, die oft durch eine Lücke
in der Kette der MTAs (IP-Adresse) bzw. der Zeitstempel erkannt werden können.
```



```
SMTP-Dialog-Beispiel: cip5010 gibt Mail an henry weiter
-------------------------------------------------------
(Zeilen mit "-": eigene Eingabe; Zeilen mit "=>": Antwort von henry)

   - telnet henry smtp                            Verbindung aufbauen
   => 220 cipgate.cfd ESMTP

   - ehlo cip5010.cfd                             "Hallo" sagen
   => 250 cipgate.cfd

   - mail from: demo@henry.pi.tu-berlin.de        Sender angeben        |
   => 250 Ok                                                            |
                                                                        => Envelope
   - rcpt to: oergel@pi.tu-berlin.de              Empfänger angeben     |
   => 250 Ok                                                            |

   - data                                         Mail übermitteln
   => 354 End data with <CR><LF>.<CR><LF>

   - From: Demo User <demo@henry.pi.tu-berlin.de>                       |
   - To: Lars Oergel <oergel@pi.tu-berlin.de>     Nachrichtentext       => Header
   - Subject: Demonstration                                             |
   -                                              (Trenner: Leerzeile)
   - Hallo Lars                                                         => Body
   - .                                            Ende: "."
   => 250 Ok: queued as 8CB7EFB75

   - quit                                         Verbindung trennen
   => Bye

Beachte:
  - Trennung Envelope und Rest der Mail
  - keine Authentifizierung (Feststellung des Absenders) nötig/möglich
  - Mailinhalt (inkl. Header) für Zustellung völlig irrelevant!

Beispiel-Header mit Received-Zeilen
-----------------------------------

From demo@henry.pi.tu-berlin.de  Thu Jan  2 04:18:16 2003
Return-Path: <demo@henry.pi.tu-berlin.de>
Received: from mail.zrz.tu-berlin.de (mail.zrz.TU-Berlin.DE [130.149.4.15])
        by prandtl.pi.tu-berlin.de (sendmail 8.12.3) with ESMTP id h023IGtj01090
        for <oergel@prandtl.pi.tu-berlin.de>; Thu, 2 Jan 2003 04:18:16 +0100
Received: from mail.zrz.tu-berlin.de ([130.149.4.15])
          by mail.zrz.tu-berlin.de with esmtp (exim-3.36)
          for <oergel@prandtl.pi.tu-berlin.de>
          id 18Tvs3-0004Hz-00; Thu, 02 Jan 2003 04:18:11 +0100
Received: from mail.zrz.TU-Berlin.DE ([130.149.4.15])
        by mail.zrz.TU-Berlin.DE (MailMonitor for SMTP v1.2.0 ) ;
        Thu, 2 Jan 2003 04:18:11 +0100 (CET)
Received: from henry.pi.tu-berlin.de ([130.149.47.15] helo=cipgate.cfd)
          by mail.zrz.tu-berlin.de with esmtp (exim-3.36)
          for <oergel@pi.tu-berlin.de>
          id 18Tvrt-0004Gl-00; Thu, 02 Jan 2003 04:18:01 +0100
Received: from cip5010.cfd (cip5010.cfd [192.168.2.10])
        by cipgate.cfd (Postfix) with ESMTP id 8CB7EFB75
        for <oergel@pi.tu-berlin.de>; Thu,  2 Jan 2003 04:17:20 +0100 (CET)
From: Demo User <demo@henry.pi.tu-berlin.de>
To: Lars Oergel <oergel@pi.tu-berlin.de>
Subject: Demonstration
Message-Id: <20030102031720.8CB7EFB75@cipgate.cfd>
Date: Thu,  2 Jan 2003 04:17:20 +0100 (CET)
```