

706.088 INFORMATIK 1

DAS INTERNET

WIEDERHOLUNG

- › Module in Python
- › PyPi (pip install)
- › import myModule
- › Fehlerbehandlung: Exceptions

EXCEPTION OBJEKT

- › Enthält Attribute und Methoden (Funktionen) zur Klassifizierung des Fehlers
- › Eigene Exceptions nur von Exception ableiten

```
>>> e = Exception("My generic error")
>>> e.args
('My generic error',)
>>> class MyException(Exception):
...     """Custom Exception raised in special cases of..."""
...     pass
>>> e = MyException("very severe error in...")
>>> e.__doc__
'Custom Exception raised in special cases of...'
>>> e.args
('very severe error in...',)
```

TRY - EXCEPT - ELSE - FINALLY

```
try:  
    print("all good")  
except NameError:  
    print("Undefined vars found")  
except:  
    print("Don't know this error!")  
    raise  
else:  
    print("everything is fine")  
finally:  
    print("cleaning up - always")
```

ASSERT

- › Setzt Bedingung, die, wenn falsch, zu einer Exception führt.
- › Nur zur Entwicklung sinnvoll.
- › Nur mit `__debug__ == True` aktiv
- › Wird mit `python3 -O` deaktiviert (`__debug__ = False`)

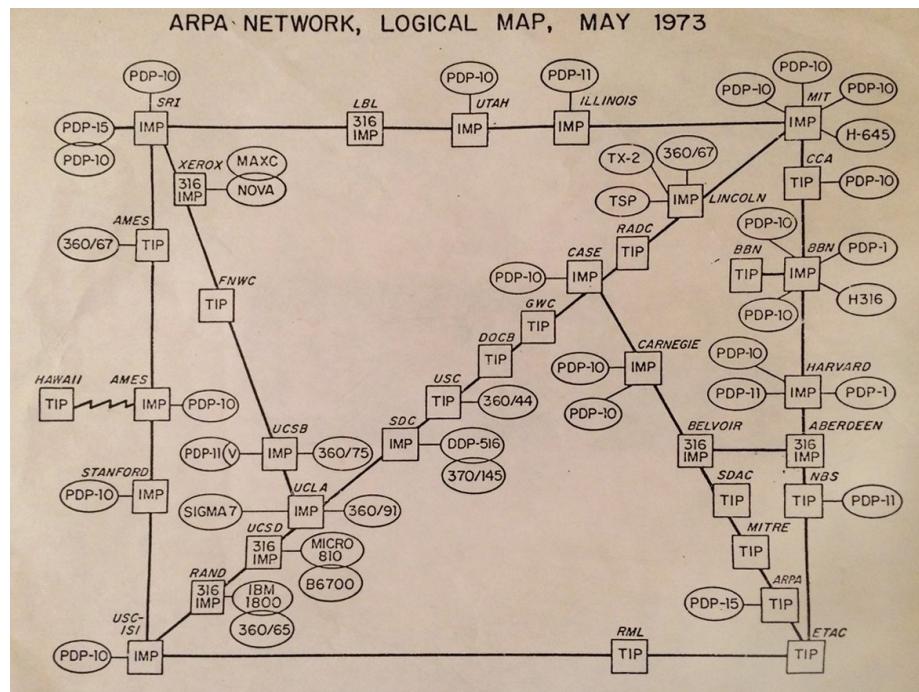


2 . 1

GESCHICHTE DES INTERNET

- › 1851: erstes Telegraphenkabel zwischen Paris und London
- › ~1866: Transatlantik-Kabel
- › 1969: ARPANET erster Vorläufer des Internet
- › 1971: **15 Knoten** sind an das ARPANET angeschlossen
- › 1987: neues Backbone für das ARPANET (27.000 Knoten)

ARPANET 1973



By ARPANET - ARPANET, Public Domain, [Link](#)

PROTOKOLL MODELLE

- › Protokoll Modelle helfen die Protokolle des Internet zu strukturieren
- › Sie sammeln die Protokolle, die verwendet werden um Daten zu übertragen
- › Sie werden in Schichten eingeteilt und zugeordnet
- › Wir sehen uns 2 Modelle genauer an:
 - » OSI-Modell
 - » DoD-Modell

OSI-MODELL

"Open Systems Interconnection Model"

Layer Nr	Name	engl. Name
1.	Bitübertragungsschicht	(Physical Layer)
2.	Sicherungsschicht	(Data Link Layer)
3.	Vermittlungsschicht	(Network Layer)
4.	Transportschicht	(Transport Layer)
5.	Sitzungsschicht	(Session Layer)
6.	Darstellungsschicht	(Presentation Layer)
7.	Anwendungsschicht	(Application Layer)

OSI-MODELL

- › Layer 1: stellt physische Übertragung von Bits zur Verfügung (Kabel, Stecker, Hub)



Public Domain, [Link](#)

OSI-MODELL

- › Layer 2: gewährleistet fehlerfreie Übertragung (IEEE 802.2, ARP, LLC, MAC, Switch, Bridge)



By [Dirk Vorderstraße - Gigabit LAN Netzwerk Switch, CC BY 2.0, Link](#)

OSI-MODELL

- › Layer 3: Weiterleitung von Datenpaketen (IP, Router, ICMP)
- › beinhaltet Weg-Suche-Mechanismen zwischen Netzwerkknoten

OSI-MODELL

- › Layer 4: Segmentierung der Daten, Stauvermeidung (TCP, UDP)
 - » TCP (Transmission Control Protocol):
 - › Fehler in der Datenübertragung werden erkannt
 - › weit verbreitet, zuverlässig
 - » UDP (User Datagram Protocol):
 - › Verbindungsloses Protokoll
 - › nur Adressierung keine Zustellgarantie
 - › schnellere Übertragung, da kein Garantie-Overhead

OSI-LAYER-MODELL

- › Layer 5: Dient der Wiederaufnahme der Datenübertragung nach Ausfall der Verbindung
- › Layer 6: Datenkomprimierung, Verschlüsselung
- › Layer 7: Dienste und Anwendungen (Web, E-Mail, IRC, Jabber, etc.)
- › Layer 1 & 2 und 5-7 werden in anderen Modellen zu einem Layer vereinfacht.

DOD-MODELL

auch bekannt als "Internet protocol suite"

Nr	Name	engl. Name	OSI Layer
1	Netzzugriff	(Network access)	1,2
2	Internet	(Internet)	3
3	Transport	(Host-to-Host)	4
4	Anwendung	(Process)	5,6,7

DOD-MODELL

benannt nach dem "Department of Defense"

- › Layer 1: löst das Problem der Datenübertragung:
Kommunikation via zB Ethernet, WLAN (ARP, MAC)
- › Layer 2: Netzwerkweite Adressierung (IPv4, IPv6, IPSec,
ICMPv6)
- › Layer 3: Verbindung 2er Prozesse auf verschiedenen
Systemen (TCP, UDP)
- › Layer 4: Definiert Aufbau der eigentlichen Nachricht
(SMTP, HTTP, DNS, DHCP etc)

ARTEN PAKETE ZU VERSCHICKEN

- › **verbindungslos**: keine Kontrolle über Zustellung (wie zB Brief)
- › **bestätigen** aber verbindungslos: Kontrolle über Empfang (Einschreiben mit Rückschein)
- › **verbindungsorientiert**: Kontrollinformation wird immer übertragen, auch wenn keine Daten geschickt werden (Pers. Gespräch, Gesichtsausdruck)
- › UDP/TCP: **UDP**: schnell und verbindungslos; **TCP**: zuverlässige Übertragung von Daten

ADRESSIERUNG

IPv4

- › IPv4: 4 8-bit Blöcke (32bit), Schreibweise mit Punkt getrennt, dezimal
 - » 127.0.0.1, 129.27.200.1
 - » 2^{32} Adressen = 4.294.967.296 Geräte
- › Netze: A: 255.0.0.0/8; B: 255.255.0.0/16;
C: 255.255.255.0/24
- › Private: z.B. 192.168.0.0/16; 10.0.0.0/8

ADRESSIERUNG

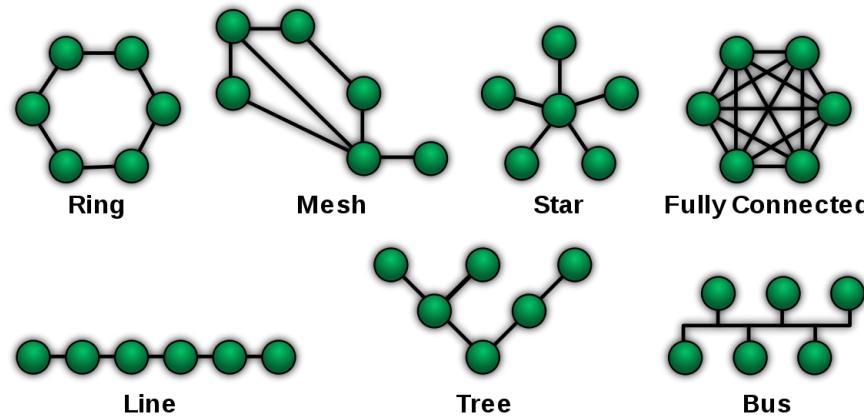
IPV6

- › IPv6: 8 16-bit Blöcke (128bit), Schreibweise mit Doppelpunkt getrennt in Hex
 - » 2a00:1450:4001:813::200e,
2001:0db8:0000:0042:0000:8a2e:0370:7334
 - » 2^{128} Adressen = $3.4 \cdot 10^{38}$
- › kein Broadcast wie IPv4, aber Multicast

IPV6

- › führende Nullen können weggelassen werden, einmal können aufeinander folgende Null-Blöcke mit '::' abgekürzt werden
 - » Bsp: 2001:0db8:0000:0000:0000:ff00:0042:8329 -> 2001:db8:0:0:0:ff00:42:8329 -> 2001:db8::ff00:42:8329
- › Localhost: '::1'
- › ICMPv6 und DNS ist für IPv6 besonders wichtig

NETZWERKTOPOLOGIEN



By NetworkTopologies.png: Maksim derivative work: Malyszkz (talk) - NetworkTopologies.png, Public Domain, Link

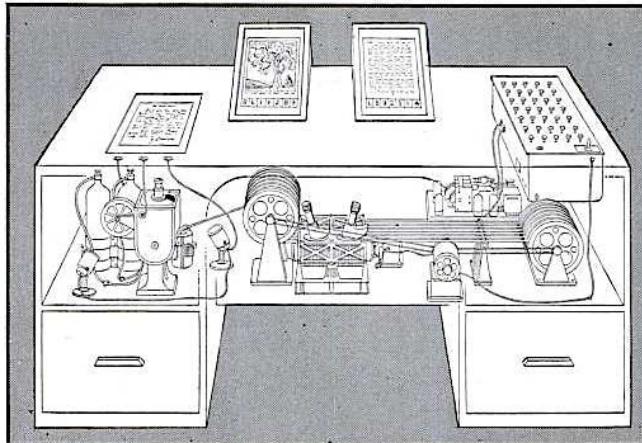
ANWENDUNGEN IM INTERNET

(via TCP/UDP)

- › Das Web (HTTP): Port 80
 - » Web mit TLS/SSL: Port 443
- › E-Mail: TCP Port 25
- › SSH: TCP Port 22
- › FTP: TCP Port 20/21
- › NTP: UDP Port 123
- › DNS: UDP/TCP Port 53

WEB - (THEORETISCHE) VORLÄUFER

› Memex (1945), in 'As we may think', von Vannevar Bush

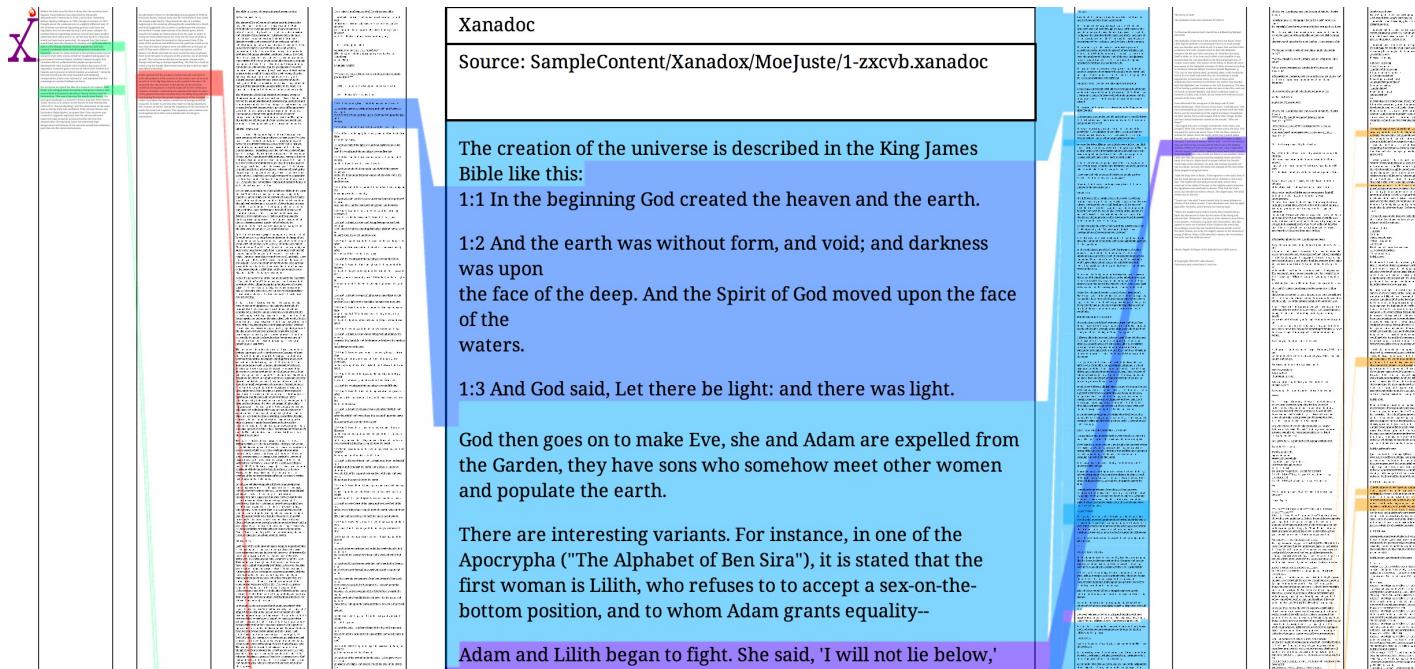


MEMEX in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicro-film filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference.

AS WE MAY THINK CONTINUED

WEB - (THEORETISCHE) VORLÄUFER

- › Hypertext, Hypermedia, Project Xanadu (1963), von Ted Nelson



WEB

- › 1989/1990 Tim Berners-Lee präsentiert das WorldWideWeb (WWW)
- › Die Idee ist es Dokumente eindeutig (URL) im Netz zu teilen und zwischen Inhalten linken zu können (Hyperlink)
- › HyperText Transfer Protocol (HTTP) war das Resultat
- › Veröffentlichung des ersten Web-Browsers und Web-Servers Dezember 1990

DEMO

WEB - HTTP

```
GET /home.html HTTP/1.1  
Host: www.example.org
```

WEB - HTTP

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=UTF-8
```

```
<html>
  <head>
    <title>Example.org - The World Wide Web</title>
  </head>
  <body>
    <p>The World Wide Web, abbreviated as WWW and commonly known ...
  </body>
</html>
```

HTTP MIT PYTHON

- › Server (nur als Development Server!):

```
$ python3 -m http.server 8000
```

- › Client:
 - » `requests` Library
 - » `urllib3`
 - » `http.client`

HTTP MIT PYTHON

Requests Library

```
>>> import requests
>>> r = requests.get('http://tugraz.at')
>>> r.status_code
200
>>> r.headers['content-type']
'text/html; charset=utf-8'
>>> r.headers['server']
'Apache'
>>> r.text[0:20]
'<!DOCTYPE html>\n<htm'
```

LINKS IM WEB

```
<a href="http://www.example.org/home.html">Example.org Homepage</a>
```

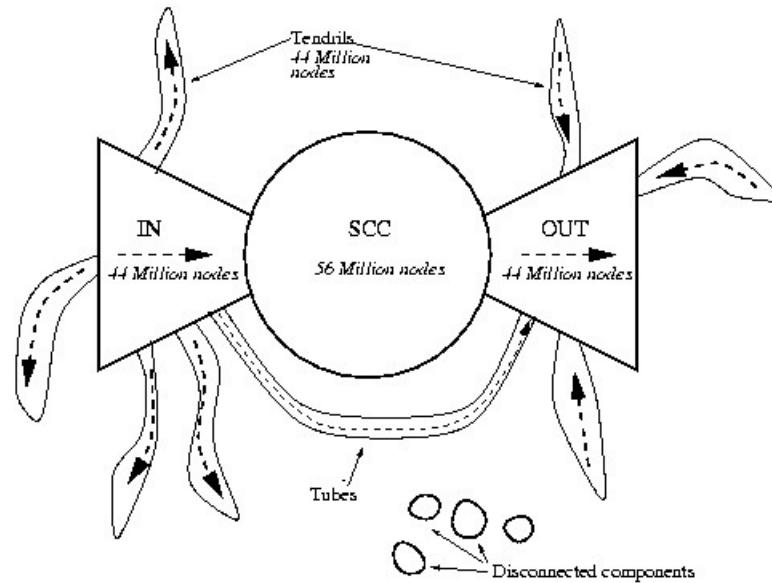
LINKS MIT PYTHON

HTML parsing

› BeautifulSoup

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(r.text,'html.parser') # Request from
for a in soup.find_all('a'):
    print(a.get('href','/'))
```

STRUKTUR DES WEB



HYPER-G - EINE KONKURRENZ AUS GRAZ

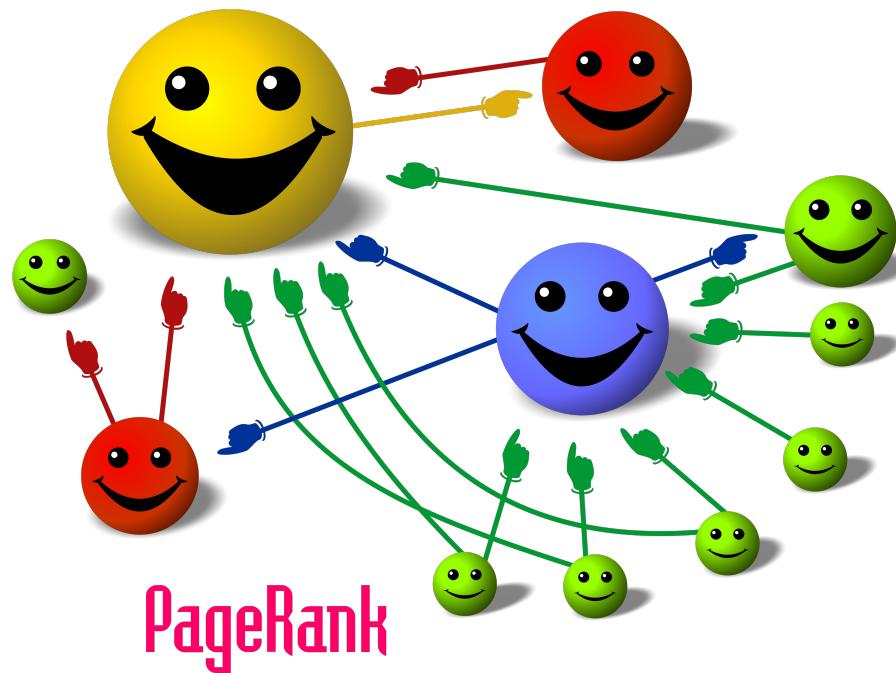
- › Ein Link im Web kann 'gebrochen' werden
 - » Wenn eine Seite nicht mehr existiert
 - » Wenn die Seite umzieht
- › Lösung:
 - » Links müssen von beiden Seiten akzeptiert werden
 - » Problem: Grosser Ressourcenbedarf und Arbeitsaufwand

WEB SUCHE

Problem: Zu viel (auch falsche) Information. Man möchte Inhalte indizieren und verfügbar machen.

- › Das Web ist wie eine Bibliothek ohne Index:
 - » Wonach indiziert man?
 - › Adresse?
 - › Inhalt?
 - › Beteiber?
 - › Netzwerkstruktur!

PAGE RANK



CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=2776582>

DNS - DOMAIN NAME SYSTEM

- › übersetzt Domain Namen in IP Adressen
- › Nameserver sind das 'Telefonbuch' des Internet
- › UDP Port 53, teilweise unterstützt von TCP (bei grossen Anfragen)

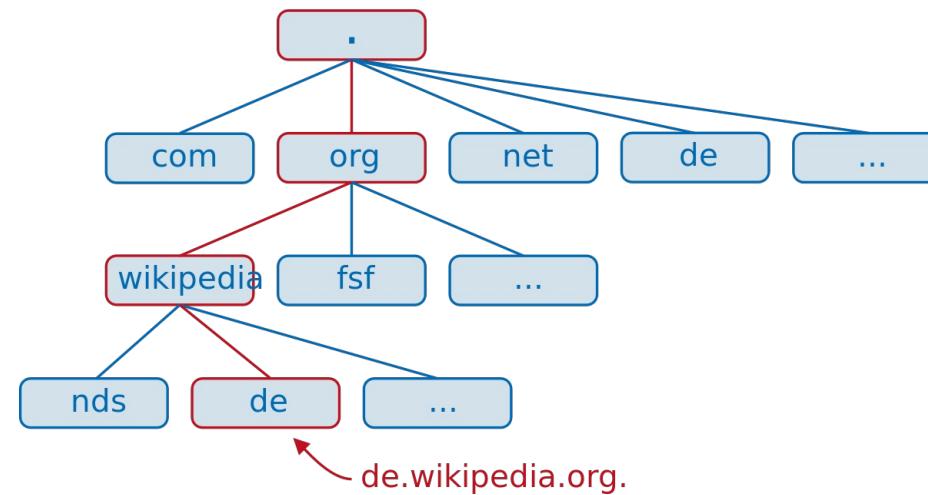
DNS - DOMAIN NAME SYSTEM

```
$ dig A tugraz.at
-----
;; ANSWER SECTION:
tugraz.at.      3600    IN     A      129.27.2.244
;; AUTHORITY SECTION:
tugraz.at.      2081    IN     NS    ns2.tu-graz.ac.at.
tugraz.at.      2081    IN     NS    ns5.univie.ac.at.
tugraz.at.      2081    IN     NS    ns10.univie.ac.at.
tugraz.at.      2081    IN     NS    ns1.tu-graz.ac.at.
-----
;; Query time: 2 msec
;; SERVER: 129.27.2.3#53(129.27.2.3)
;; WHEN: Tue Jan 10 19:25:55 CET 2017
;; MSG SIZE  rcvd: 265
```

DNS RECORD TYPES

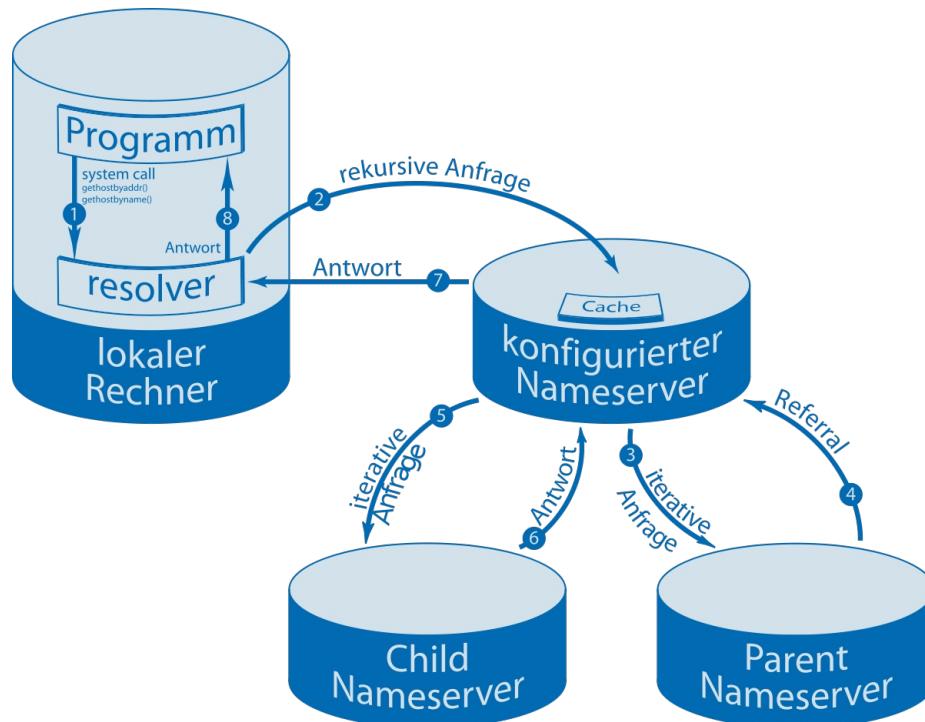
- › A: Haupteintrag IPv4
- › AAAA: Haupteintrag IPv6
- › CNAME: verweist auf anderen Namen
- › MX: Mail Exchange, verweist auf den E-Mail Server
- › NS: NameServer, Delegierung der Nameserver
untereinander
- › TXT: Textinformation (SPF, DMARC, ...)
- › SRV: Service Locator, Eintrag zu angebotenen Diensten
(zB XMPP)

DNS



CC BY-SA 2.5, Link

DNS ANFRAGE

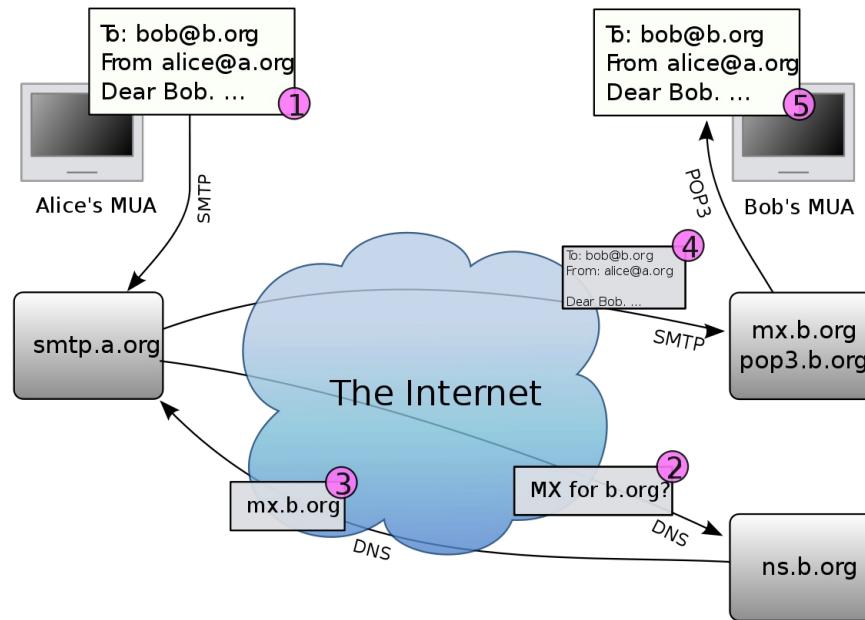


Von Hank van Helvete - Eigenes Werk, CC BY-SA 2.5, Link

E-MAIL

- › seit 1971 (erste E-Mail im ARPANET)
- › hostbasiertes Nachrichtensystem
- › E-Mail Adresse repräsentiert ein Postfach
 - » user@host

E-MAIL



By Yzmo at the English language Wikipedia, CC BY-SA 3.0, Link

FTP

File Transfer Protocol

- › Protokoll zur Übertragung von Daten zu Servern.
- › Meist für Webseiten genutzt.
- › Unverschlüsselt 
- › Port 20 (data port), Port 21 (command port)
- › Verschieden Modi für Datenformate (ASCII, binary, ...)

SSH

Secure Shell, zB OpenSSH

- › verschlüsselte Verbindung zum:
 - » Ausführen von Kommandos (ssh)
 - » Übertragen von Dateien (scp, rsync, SFTP)
 - » Passwortlosen Login
 - » Tunneln von Verbindungen (ssh -L, ssh -R)
 - » Verbinden via SOCKS Proxy (ssh -D)
 - » Einbinden des Filesystems (sshfs)

FRAGEN?

5 . 22