

7

OOP – Linux

Ausgewählte Themen zu Web-Servern

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-dresden.de



GLIEDERUNG DER VORLESUNG

Einführung: Geschichte von Unix zu Linux

Kapitel 1: Allgemeines und Grundlagen

Kapitel 2: Arbeit mit der Kommandozeile

Kapitel 3: Boot-Vorgang und Systeminitialisierung

Kapitel 4: Ausgewählte Themen der Systemadministration

Kapitel 5: Ausgewählte Themen der Netzwerkkonfiguration

Kapitel 6: Anwendungsentwicklung unter/für Linux

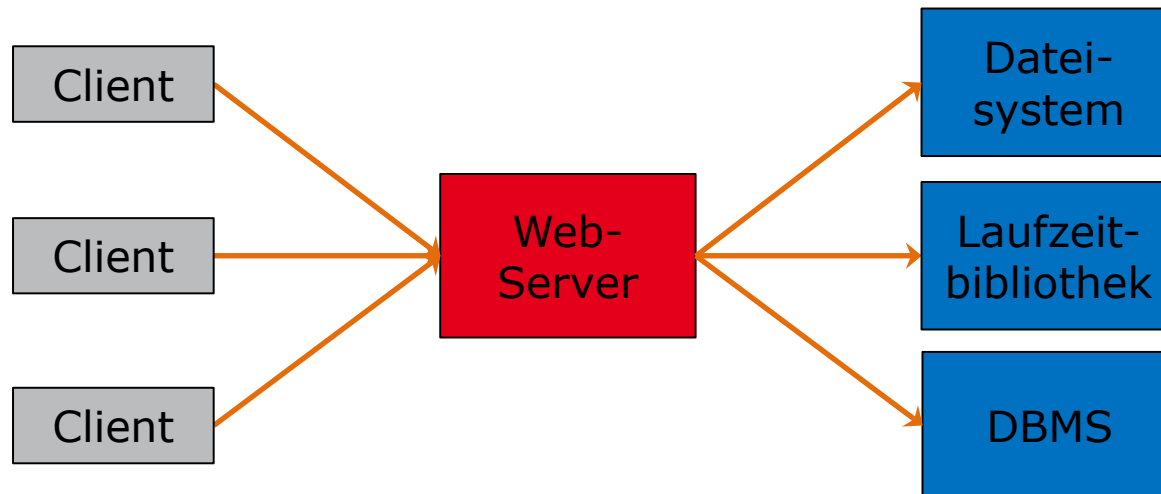
Kapitel 7: Ausgewählte Themen zu Web-Servern

INHALTE

- WWW
- DNS
- E-Mail

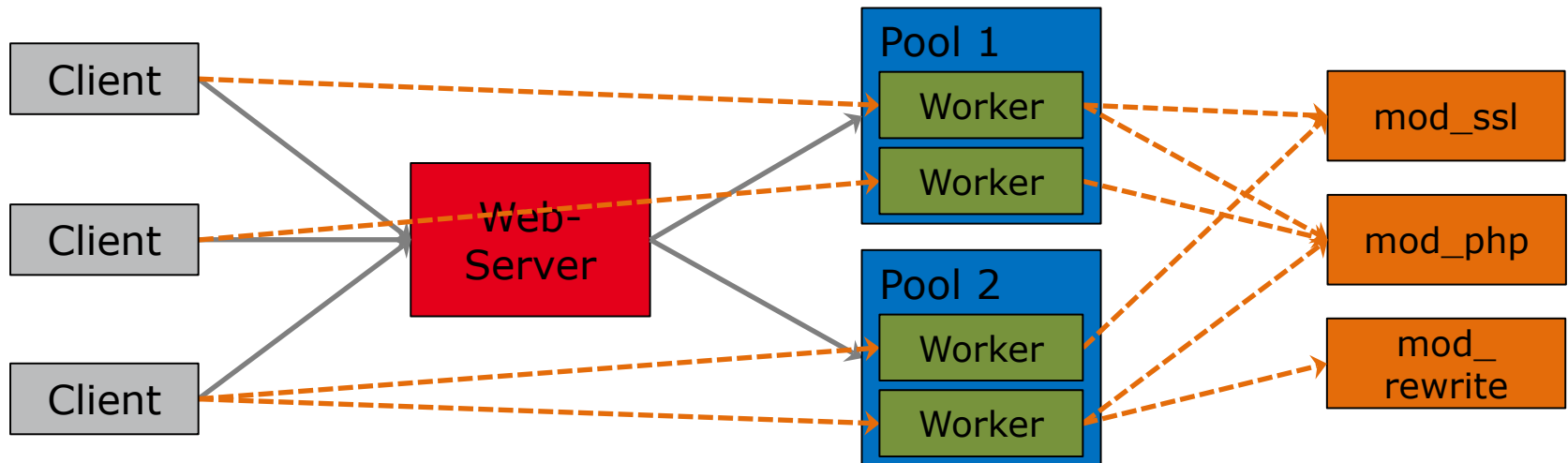
WEB-SERVER

- Anfragebasierte Bereitstellung von Ressourcen
 - Dateien
 - strukturierte Informationen
 - (partielle) Substitutionen
- i.d.R. viele parallele, teilweise lange offene Verbindungen
 - große Dateien
 - TCP Connection Keep Alive
 - WebSockets
 - ...



WEB-SERVER – APACHE HTTPD

- Vorhalten von wartenden Workern (Threads und Prozesse)
 - jede TCP-Verbindung wird einem Worker zugewiesen
 - bei Erschöpfung eines Worker-Pools wird neuer Pool spawned bis maximale Speicher- oder Prozessorauslastung erreicht
- organisiert in Web-Server-Kern (httpd) und eingebundene Module
 - HTTPs
 - WebSockets
 - PHP
 - ...



WEB-SERVER – APACHE HTTPD

- Aktivierung/Deaktivierung von Modulen via Softlink von `/etc/apache2/mods-available` nach `/etc/apache2/mods-enabled`
 - niemals von Hand setzen!
 - Hilfsprogramme `a2enmod` und `a2dismod`
- grundlegende Definition des zu bedienenden Ports via `/etc/apache2/ports.conf`

```
NameVirtualHost *:7080 # für virtuelle Hosts
Listen 7080           # Standard-Port

<IfModule mod_ssl.c> # für SSL-Verschlüsselung
    NameVirtualHost *:7443
    Listen 7443
</IfModule>

<IfModule mod_gnutls.c> # für TLS-Verschlüsselung
    NameVirtualHost *:7443
    Listen 7443
</IfModule>
```

WEB-SERVER – APACHE HTTPD – APACHE2.CONF

```
# Allgemeine Struktur
# /etc/apache2/
# |-- apache2.conf <-- globale Konfiguration
# |   |-- ports.conf
# |-- mods-enabled
# |   |-- *.load
# |   |-- *.conf <-- Modulkonfiguration
# |-- conf.d
# |   |-- *
# |-- sites-enabled
# |   |-- * <-- Virtual-Host-spezifische Konfiguration

ServerRoot "/var/www/vhosts/default" # Standardverzeichnis des Servers

# ...

# Verbindungsparameter wie z.B. TCP Connection Keep Alive
Timeout 300
KeepAlive On
MaxKeepAliveRequests 127
KeepAliveTimeout 30
```

WEB-SERVER – APACHE HTTPD – APACHE2.CONF

```
# Worker-Pool
<IfModule mpm_prefork_module>
    StartServers          16
    MinSpareServers       32
    MaxSpareServers       48
    MaxClients            127
    MaxRequestsPerChild   1023
</IfModule>

<IfModule mpm_worker_module>
    # weitesgehend analog, aber Abweichungen möglich
    StartServers          7
    # ...
    ThreadLimit           63
    ThreadsPerChild       15
    # ...
</IfModule>

<IfModule mpm_event_module>
    # analog
</IfModule>

# Rechte im System (hier über Environment gesetzt)
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

WEB-SERVER – APACHE HTTPD – APACHE2.CONF

```
# ...

# Fehler protokollieren
ErrorLog ${APACHE_LOG_DIR}/error.log
LogLevel crit

# Einbinden der Module
Include mods-enabled/*.load
Include mods-enabled/*.conf

# Festlegen der Ports, auf die geantwortet werden soll
Include ports.conf

# ...

# Virtual-Host-Konfiguration einbinden
Include sites-enabled/

# ...

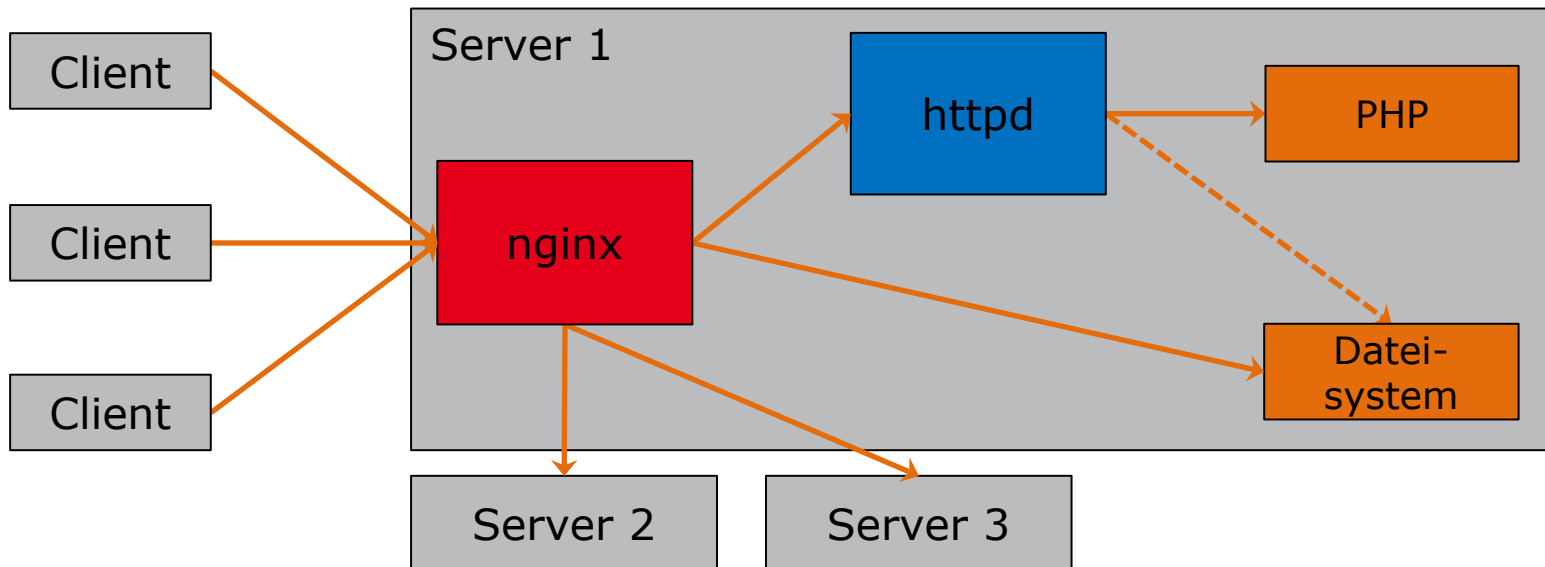
# Sonstige Direktiven, z.B. für PHP
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

WEB-SERVER – APACHE HTTPD – VIRTUELLE HOSTS

```
<VirtualHost ba-dresden.de:7080>
  ServerAdmin admin@ba-dresden.de
  DocumentRoot /var/www/vhosts/ba-dresden.de/htdocs/
  <Directory />
    Options FollowSymLinks
    AllowOverride None # keine .htaccess erlauben
  </Directory>
  <Directory /geheim/>
    Options Indexes FollowSymLinks MultiViews
    Order deny,allow
    allow from 195.37.37.0/24
  </Directory>
  # Laufzeitbibliotheken wie PHP müssen als CGI oder FCGI bereitstehen
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>
  # Virtual-Host-spezifische Fehlerprotokollierung
  ErrorLog ${APACHE_LOG_DIR}/ba-dresden.de/error.log
  LogLevel crit
</VirtualHost>
```

WEB-SERVER – NGINX

- Apache httpd in der Regel bei hochfrequentierten Webseiten ineffizient
 - hoher Speicherbedarf
 - hohe Systemlast
- vorgeschalteter Proxy: nginx
 - bedient „einfache“ Anfragen
 - statische Inhalte wie Bild-, CSS- oder JS-Dateien
 - dynamisch zu erzeugende Inhalte kommen weiterhin von httpd
 - Load-Balancing



WEB-SERVER – NGINX – NGINX.CONF

Konfigurationsstruktur ist ähnlich der von httpd

```
user nginx;
worker_processes 8;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 30;
    # weitere Konfiguration, bspw. Kompression oder Laufzeitbibliothek
    gzip on;
    gzip_disable "MSIE [1-6]\.(?!.*SV1)";
    server_tokens off;
    include /etc/nginx/conf.d/*.conf;
    fastcgi_buffers 8 16k;
    fastcgi_buffer_size 32k;
}
```

WEB-SERVER – NGINX – VIRTUELLE HOSTS

```
server {
    listen 443 ssl http2;
    server_name ba-dresden.de;
    server_name www.ba-dresden.de;

    ssl_certificate      /etc/dfn-crypt/ba-dresden.de/cert.pem;
    ssl_certificate_key  /etc/dfn-crypt/ba-dresden.de/privkey.pem;

    client_max_body_size 128m;
    root "/var/www/vhosts/ba-dresden.de/htdocs";

    if ($host ~* ^ba-dresden.de$) {
        rewrite ^(.*)$ https://www.ba-dresden.de$1 permanent;
    }

    location / {
        proxy_pass https://195.37.37.47:7443;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        access_log off; # wird von httpd protokolliert
    }
}
```

WEB-SERVER – NGINX – VIRTUELLE HOSTS

```
location @fallback {
    proxy_pass http://195.37.37.47:7080;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    access_log off;
}

location ~ ^/(.*\.(ac3|avi|bmp|bz2|css|cue|dat|doc|docx|dts|exe|flv|gif|gz|
    htm|html|ico|img|iso|jpeg|jpg|js|mkv|mp3|mp4|mpeg|mpg|ogg|pdf|png|
    ppt|pptx|qt|rar|rm|swf|tar|tgz|txt|wav|xls|xlsx|zip))$ {
    try_files $uri @fallback;
}

location ~ ^/not-httpd/ {
    try_files /not-httpd/$uri
}

include "/var/www/vhosts/system/ba-dresden.de/conf/vhost_nginx.conf";
}

server {
    listen 80;
    server_name ba-dresden.de;
    server_name www.ba-dresden.de;
    rewrite ^(.*)$ https://www.ba-dresden.de$1 permanent;
}
```

DOMAIN NAME SYSTEM

- kein eigener DNS-Server zum Betrieb eines Web-Servers notwendig
 - Namensauflösung geschieht durch **named**
 - oder statisch in **/etc/resolv.conf**
- oft agieren zentrale Server aus Kostengründen gleichzeitig auch als
 - LDAP-Server
 - Mail-Server
 - SMB-Server
 - ...
 - und eben auch als DNS-Server zur
 - Namensauflösung
 - Zonenverwaltung

DOMAIN NAME SYSTEM – BIND

- 1984: Berkeley Internet Name Domain vorgestellt (Paper)
- 1988: von der UCB mit BSD 4.3 veröffentlicht und durch CSRG der UCB verwaltet
- Anfang 1990er: durch Vixie Enterprises verwaltet und in 1994 in Internet Systems Consortium (eine US 501(c)(3)) ausgegründet
- 2000: vollständige Reimplementierung durch Nominum, Inc.
 - DNSSEC- und IPv6-Unterstützung
 - Split-Finanzierung durch ISC, Open-Source-Community und US-Militär
- 2009: Fork von BIND durch ISC → BIND10

Achtung! Bind steht unter Mozilla-Lizenz, nicht GPL!

Ist dennoch sehr weit verbreitet (vornehmlich BIND9).

DOMAIN NAME SYSTEM – BIND

Konfiguration der wichtigsten Zoneninformationen in
[/etc/bind/bind.conf](#) oder [/etc/bind/*.conf.*](#)

```
zone "." {  
    type hint;  
    file "named.root";  
    // Verweis auf ftp://ftp.rs.internic.net/domain/named.root  
};  
// ...  
zone "ba-dresden.de" {  
    type master;  
    file "ba-dresden.de";  
    // muss in /etc/domain/ba-dresden.de beschrieben werden!  
};  
zone "37.37.195.in-addr.arpa" {  
    type master;  
    file "195.37.37";  
};  
zone "kirk.ba-dresden.de" {  
    type master;  
    file "195.37.37.18";  
};
```

DOMAIN NAME SYSTEM – NAMED

Konfiguration der Auflösungsinformationen in [/etc/named.boot](#)

```
directory /etc/domain

cache .named.root
; wieder Verweis auf ftp://ftp.rs.internic.net/domain/named.root

primary localhost localhost
primary 0.0.127.in-addr.arpa 127.0.0

primary ba-dresden.de ba-dresden.de
; wieder Datei /etc/domain/ba-dresden.de

primary 37.37.195.in-addr.arpa 195.37.37
; wieder Datei /etc/domain/195.37.37
```

DOMAIN NAME SYSTEM – LOCALHOST-INFO

Konfiguration des Localhosts in [/etc/domain/localhost](#)

```
localhost. SOA kirk.ba-dresden.de. sachse.ba-dresden.de. (  
    2017072602    ; eindeutige Seriennummer  
    21600         ; Aktualisierungsintervall in s  
    7200          ; Wiederholungsintervall in s  
    1814400       ; Ablaufzeit in s  
    86400         ; min TTL in s  
)  
  
localhost. NS   kirk.ba-dresden.de.  
  
localhost. A    127.0.0.1
```

Analoges in [/etc/domain/127.0.0](#)

DOMAIN NAME SYSTEM – DOMAIN-INFO

Konfiguration der eigentlichen Zone in [/etc/domain/ba-dresden.de](#)

```
ba-dresden.de.  SOA  kirk.ba-dresden.de.  sachse.ba-dresden.de.
(
    2017072602
    21600
    7200
    1814400
    86400
)

ba-dresden.de.      NS  kirk.ba-dresden.de.
ba-dresden.de.      NS  deneb.dfn.de.
ba-dresden.de.      MX  10 egon2.rz.ba-dresden.de.
www.ba-dresden.de.  A   195.37.37.47
kirk.ba-dresden.de. A   195.37.37.18
egon2.rz.ba-dresden.de. A 195.37.37.12
```

DOMAIN NAME SYSTEM – REVERSE-INFO

Konfiguration der reversen Informationen in [/etc/domain/195.37.37](#)

```
37.37.195.in-addr.arpa. SOA kirk.ba-dresden.de. sachse.ba-dresden.de. (  
    2017072602  
    21600  
    7200  
    1814400  
    86400  
)  
  
37.37.195.in-addr.arpa.      NS      kirk.ba-dresden.de.  
37.37.195.in-addr.arpa.      NS      deneb.dfn.de.  
12.37.37.195.in-addr.arpa.   PTR      egon2.rz.ba-dresden.de  
18.37.37.195.in-addr.arpa.   PTR      kirk.ba-dresden.de.  
47.37.37.195.in-addr.arpa.   PTR      www.ba-dresden.de.
```

Schlüssel für DNSSEC notwendig → [/etc/bind/bind.keys](#)

DNS-Daten für ba-dresden.de: <http://viewdns.info/dnsreport/?domain=ba-dresden.de>

Informationen zur Organisation der Root-Zone: <ftp://ftp.rs.internic.net/domain>

E-MAIL

Diverse Komponenten für E-Mail notwendig

- Mail Transfer Agent (MTA): Eigentlicher SMTP-Server
 - Postfix
 - Exim
 - Sendmail
 - ...
- Mail User Agent (MUA): Client-seitige Software
- SMTP-Server müssen mit einander sprechen
 - Portfreigaben (Netzwerkzugriff)
 - Daemonen (Server müssen opportunistisch warten)

E-MAIL

Diverse Protokolle und Mechanismen an E-Mail-Austausch beteiligt

- Post-Office Protocol (POP3):
Kommunikationsprotokoll mit dem MUA E-Mail vom MTA abholen kann
- Interactive Message Access Protocol (IMAP):
ähnlich POP3, aber E-Mail verbleiben auf dem Server bis sie explizit gelöscht werden (Speicherbedarf auf Server!)
- Simple Authentication and Security Layer (SASL):
Nutzer-Authentifikation (unabhängig von Verschlüsselung!)
- Transport Layer Security (TLS):
zur Verschlüsselung der Kommunikation

E-MAIL

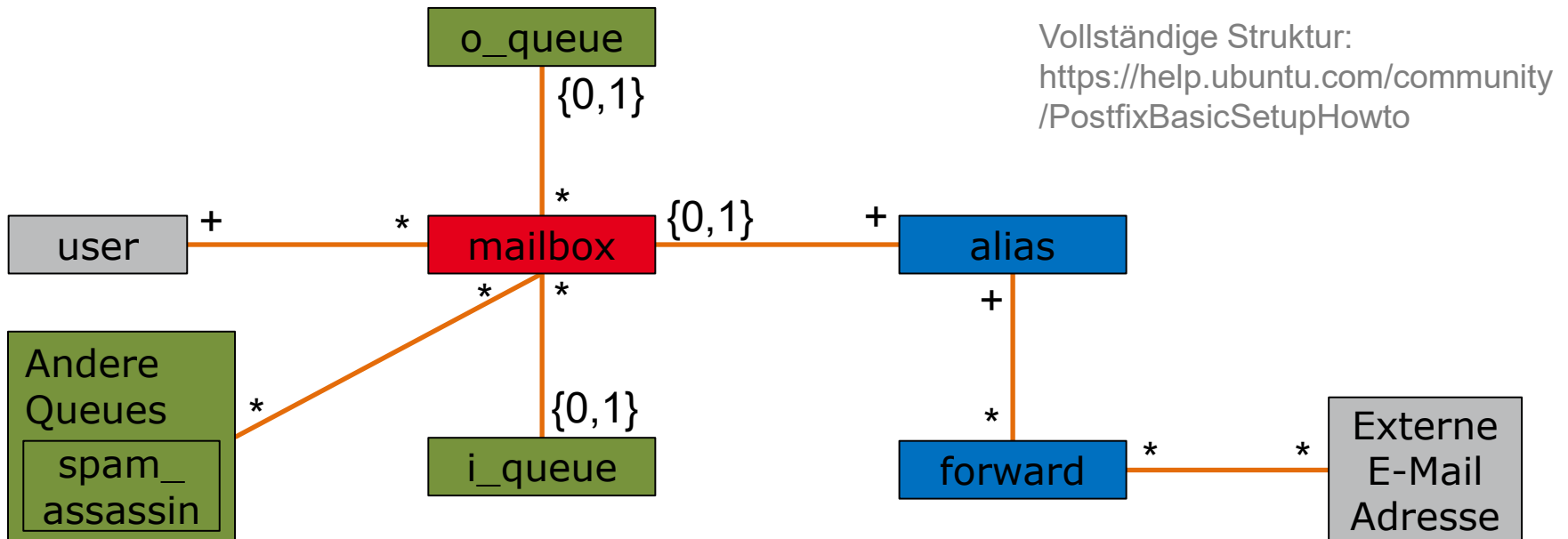
In der Regel benötigte Portfreigaben:

- 25 (SMTP)
- 110 (POP3)
- 143 (IMAP)
- 465 (SMTPs)
- 995 (POP3s)
- 993 (IMAPs)

Proprietäre Lösungen wie Exchange arbeiten ähnlich, aber doch anders.

E-MAIL – MTA

- **user**: E-Mail-Nutzer (generiert oft jdoe1234@example.com)
- **mailbox**: Postfach in dem E-Mails liegen
- **alias**: (alternative) Adresse eines Postfaches (john.doe@example.com)
- **forward**: Weiterleitung auf mind. einen Aliase (johns@example.com)
- **queue**: Warteschlange für Verarbeitung (sowohl Ein- als auch Ausgang)



E-MAIL – POSTFIX

- weit verbreiteter MTA
- simpel umzusetzen
- nach Installation stehen drei Daemonen bereit:
 - **master**: der eigentliche MTA
 - **qmgr**: zur Verwaltung der Queues und Verteilung von E-Mails auf die Queues
 - **pickup**: Abholung von E-Mails von anderen MTA

```
user@linux$ sudo apt-get install postfix
...
user@linux$ ps ax
 9423 ? Ss  15:22 /usr/lib/postfix/master
25032 ? S    0:00 pickup -l -t fifo -u -c
 9437 ? S   12:00 qmgr -l -t fifo -u
...
user@linux$ sudo apt-get install mailutils
```

E-MAIL – POSTFIX

- Konfiguration ist sehr eng an Linux-Nutzermodell gekoppelt
- E-Mails werden von Queue im User-Verzeichnis abgelegt
`/home/user/Maildir/*`
- Aliase werden in `/etc/aliases` definiert

```
john.doe: jdoe1234
webmaster: root
```
- Forwards werden i.d.R. an Nutzerkonten gekoppelt und beim User definiert in `/home/user/.forward`

```
# johns@example.com
jdoe1234
jmiller0
john.carpenter@example.org
```
- nach jeder Änderung an den Aliasen manuelles Update notwendig:

```
user@linux$ sudo newaliases
```

E-MAIL – POSTFIX

E-Mails werden in der Regel im Klartext vom MTA (zwischen)gespeichert, bspw. in `/var/mail/*`

```
# /var/mail/jdoe1234
From MAILER-DAEMON Sat Sep 23 14:56:17 2017
Return-Path: <>
X-Original-To: john.doe@example.com
Delivered-To: jdoe1234@example.com
Received: by example.com (Postfix)
        id 6B7EB9E11B7; Sat, 23 Sep 2017 14:55:34 +0000 (UTC)
Date: Sat, 23 Sep 2017 14:55:34 +0000 (UTC)
From: MAILER-DAEMON@example.com (Mail Delivery System)
Subject: Test
To: john.doe@example.com
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii;
        boundary="3E7649E11B0.1419118114/example.com"
Content-Transfer-Encoding: 8bit
Message-Id: <20141220232834.6B7EB9E11B7@example.com>

This is the test mail to john.doe@example.com sent by Postfix.

--3E7649E11B0.1419118114/example.com
```

HILFSTOOLS

- Server-Verwaltung rein aus der Kommandozeile heraus oft schwierig
- viele wiederkehrende Aufgaben, wie Anlegen
 - neuer E-Mail-Adressen
 - neuer Weiterleitung
 - neuer Filter-Regeln (Queue-Anpassungen)
 - neuer virtueller Hosts
 - ...
- Abhilfe durch Server-Suites
 - Plesk, C-Panel, Webmin, Core, ...
 - oft mit grafischer Nutzeroberfläche, benötigt aber laufenden Web-Server (bspw. Plesk auf Port 8443)
 - Verwaltung weiterer Komponenten oft möglich
 - Datenbank-Server (MariaDB, MySQL, ...)
 - IPv6-Konfiguration
 - LDAP, SMB, ...

AUFGABEN

1. Installieren und konfigurieren Sie Apache **httpd**. Testen Sie, ob Sie die Standard-Seite („It works!“) abrufen können (mit Browser oder **curl** auf localhost/127.0.0.1).
2. Passen Sie das Logging von httpd so an, dass Sie im Access-Log den vollständigen User-Agent protokollieren.
3. Passen Sie die seitenspezifische Konfiguration der Standard-Seite (entweder über die Config oder eine .htaccess-Datei) so an, dass Zugriffe von unterschiedlichen User-Agents zu unterschiedlichen Dokumenten führen. (bspw. Browser → index.html; curl → blocked.html)
4. Wiederholen Sie die Aufgaben 1. bis 3. für **nginx**. Legen Sie den dafür Listening-Port von httpd auf einen anderen Port.
5. Konfigurieren Sie Ihren Server so, dass httpd dynamische Inhalte ausliefert, während nginx als Reverse-Proxy alle statischen Inhalte ausliefert. Installieren Sie dazu eine Backend für dynamische Web-Seiten, z.B. PHP.
6. Konfigurieren Sie einen (lokalen) DNS-Namen für Ihr Linux und prüfen Sie, ob Sie Ihre Webseite im Browser über die (lokale) DNS-Auflösung erreichen.