

Der MTA Exim

Erkan Yanar
erkan.yanar@t-online.de

Zusätzliche Kommentare im grün

Der MTA Exim

Einer der drei "großen" Sendmailerben

Als einziger unter der GPL

Author: Phillip Hazel

Aktuelle Version: 4.30

Themen des Vortrags

- ACL (Access Control List)
 - Mail während der Verbindung ablehnen
- Router
 - Weg festlegen
- Transports
 - Mailtransport

Idee des Vortrages

- Den Weg einer Mail nachzuzeichnen, welche von unserem MTA Exim entgegengenommen und weiterverarbeitet wird.
- Parallel betrachten wir die dazugehörigen Optionen in der Konfigurationsdatei.

Die Konfigurationsdatei

- /etc/exim/exim.conf
- Besteht aus 7 Abschnitten
- acl, router und transports haben jeweils einen Abschnitt
- Abschnitte beginnen mit
begin BEZEICHNER (z.B. begin acl)
- bis auf den ersten, dem globalen Abschnitt

Globale(runtime) Konfigurationen

- named lists
 - Definition von Listenvariablen
- Welche ACL soll wann benutzt werden

Die Konfigurationsdatei, Hauptkonfiguration named lists

Syntax:

ListenTyp ListenName = Zuweisung

ListenTypen

- domainlist

- Zielrechner

- hostlist

- Senderrechner

- : trennt mehrere Zuweisungen

- Zugriff durch vorangestelltes +

domainlist local_domains = localhost : mein.rechner.tld

hostlist relay_from_hosts = 192.168.0.2 : 10.0.0.0/8

Die Konfigurationsdatei, Hauptkonfiguration, named lists named list(2) Syntaxmöglichkeiten

■ * klassischer Wildcard

- domainlist test_domain = *.test.tld

■ ! Negation

- domainlist test_domain = ! maus.test.tld

■ / Pfadangabe zu Datei

- domainlist test_domain =/domains/in/einer/Datei

Die Konfigurationsdatei, Hauptkonfiguration, named lists

Unterschied hostlist domainlist

- domainlist meint eine Liste von Zielrechnern
- hostlist meint eine Liste von Senderrechnern
- domainlist arbeitet auf Namensbasis
- hostlist arbeitet mit IP-Adressen. Namen werden umgewandelt.

Die Konfigurationsdatei, Hauptkonfiguration

ACL

- Im globalen Abschnitt wird definiert welche ACL wann benutzt werden.

`acl_smtp_rcpt = acl_check_rcpt`

- `acl_smtp_rcpt`
 - Welche ACL soll nach dem RCPT-Kommando ausgeführt werden
- `acl_check_rcpt`
 - Name der ACL, die ausgeführt werden soll

Die Konfigurationsdatei, Hauptkonfiguration

ACL

```
220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100
HELO meine.adresse.tld
250 test.ja Hello meine.adresse.tld [192.168.0.111]
MAIL FROM: ich@meine.adresse.tld
250 OK
RCPT TO: erkan@test.ja
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
.
```

Die Konfigurationsdatei, Hauptkonfiguration ACL

220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100

HELO meine.adresse.tld

250 test.ja Hello meine.adresse.tld [192.168.0.111]

MAIL FROM: ich@meine.adresse.tld

250 OK

RCPT TO: erkan@test.ja

250 Accepted

DATA

354 Enter message, ending with "." on a line by itself

.

Die Konfigurationsdatei, Hauptkonfiguration

ACL

220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100

[HELO meine.adresse.tld](#)

250 test.ja Hello meine.adresse.tld [192.168.0.111]

MAIL FROM: ich@meine.adresse.tld

250 OK

RCPT TO: erkan@test.ja

250 Accepted

DATA

354 Enter message, ending with "." on a line by itself

.

[acl_smtp_helo](#)

acl_smtp_helo = liste_von_acls

Die Konfigurationsdatei, Hauptkonfiguration

ACL

```
220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100
HELO meine.adresse.tld
250 test.ja Hello meine.adresse.tld [192.168.0.111]
MAIL FROM: ich@meine.adresse.tld
250 OK
RCPT TO: erkan@test.ja
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
.
```

[acl_smtp_mail](#)

`acl_smtp_mail = liste_von_acls`

Die Konfigurationsdatei, Hauptkonfiguration

ACL

```
220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100
HELO meine.adresse.tld
250 test.ja Hello meine.adresse.tld [192.168.0.111]
MAIL FROM: ich@meine.adresse.tld
250 OK
RCPT TO: erkan@test.ja
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
.
```

`acl_smtp_rcpt`

```
acl_smtp_rcpt = liste_von_acls
```

Die Konfigurationsdatei, Hauptkonfiguration

ACL

```
220 test.ja ESMTP Exim 4.12 Tue, 02 Mar 2004 13:03:05 +0100
HELO meine.adresse.tld
250 test.ja Hello meine.adresse.tld [192.168.0.111]
MAIL FROM: ich@meine.adresse.tld
250 OK
RCPT TO: erkan@test.ja
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
.
```

[acl_smtp_data](#)

```
acl_smtp_data = liste_von_acls
```


Die Konfigurationsdatei, Hauptkonfiguration

ACL-Optionen

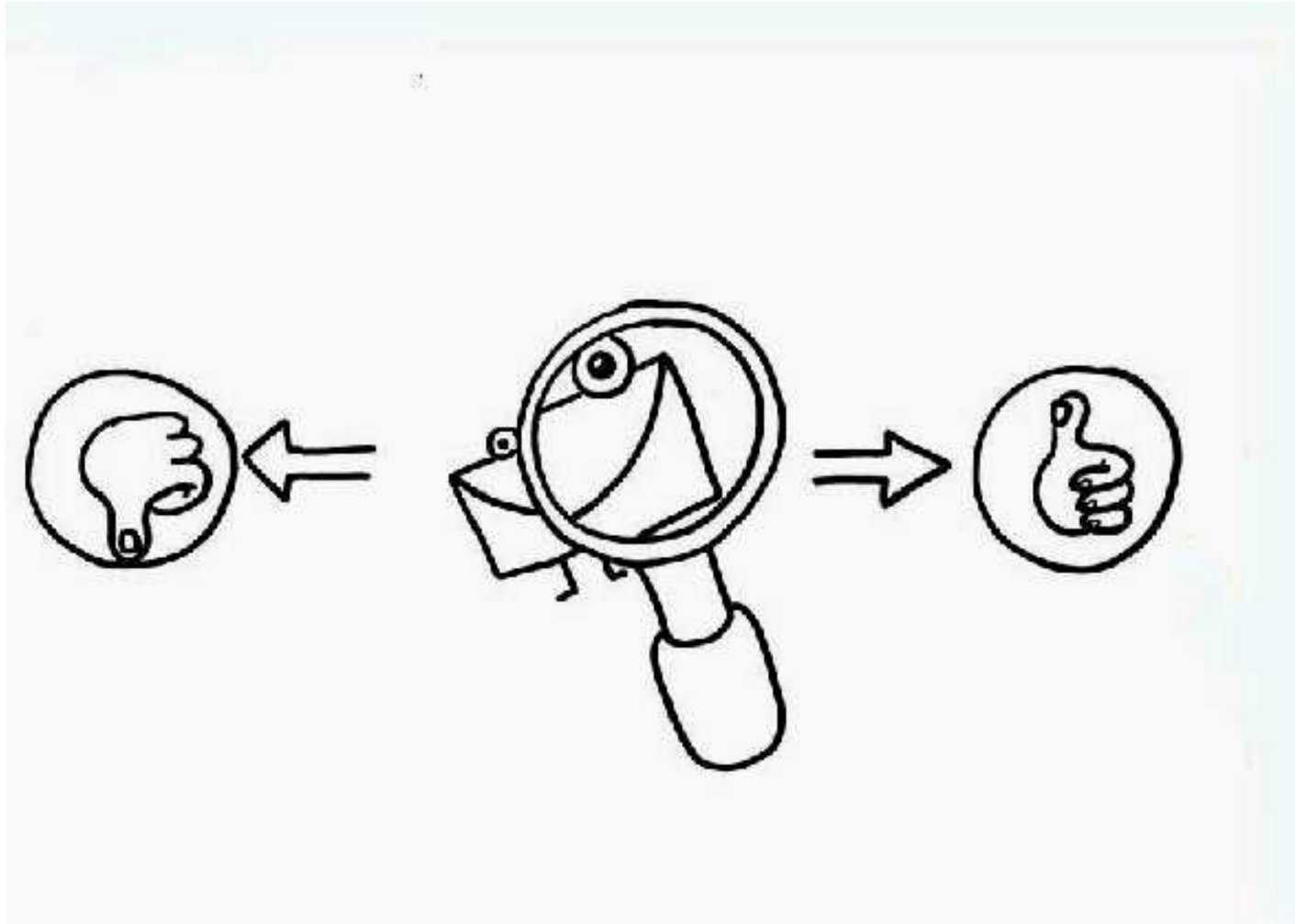
■ `acl_smtp_helo`

■ `acl_smtp_mail`

■ `acl_smtp_rcpt`

■ `acl_smtp_data`

ACL



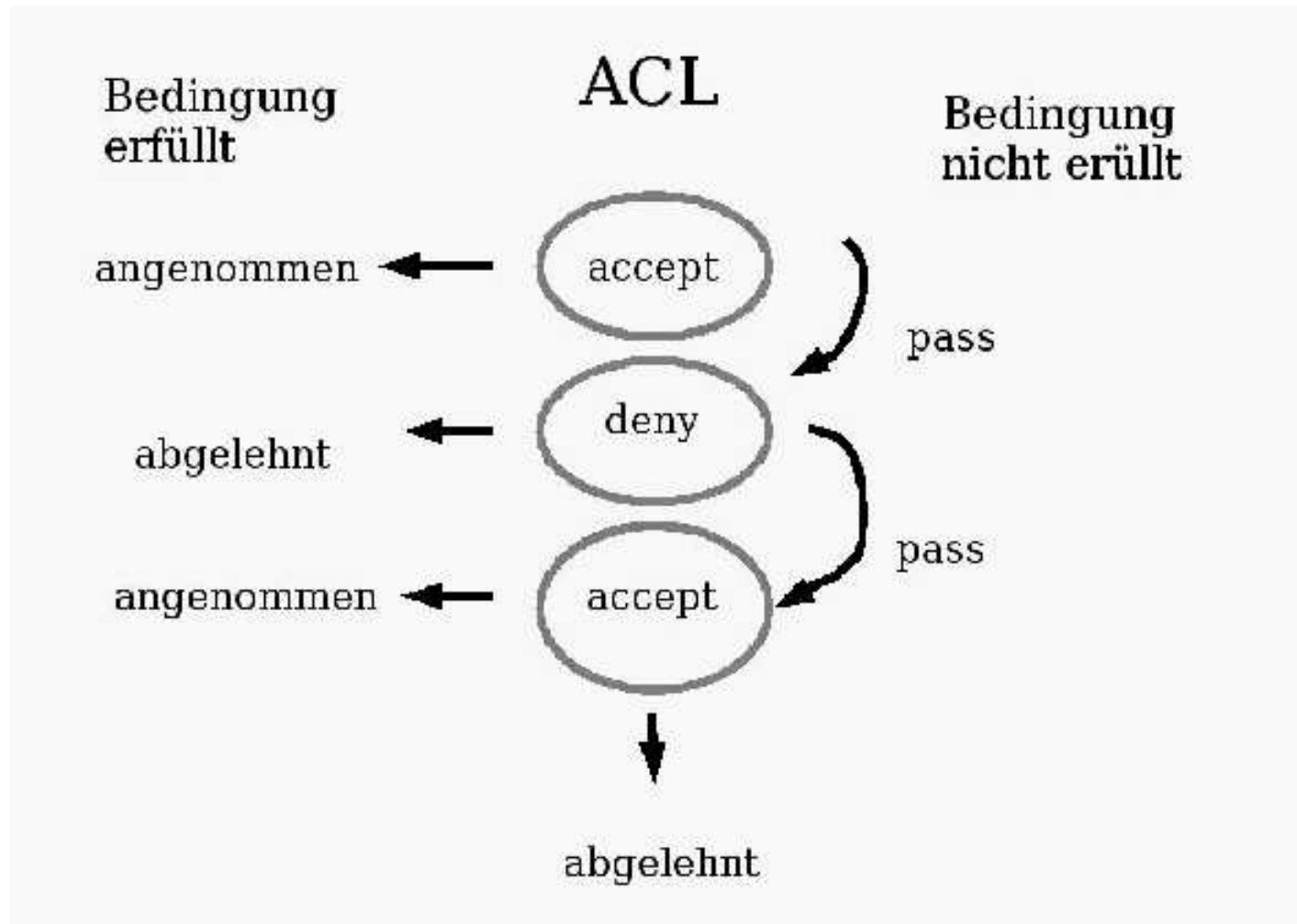
Die Konfigurationsdatei, ACL-Abschnitt

ACLs

- ACLs haben einen eigenen Abschnitt
 - begin acl
- Hier werden ACLs definiert
- ACLs können die Annahme der Mail verweigern
- ACL besteht aus
 - ACL-Namen (gefolgt von einem :)
 - Den "Verben" mit den dazu gehörigen Bedingungen

Die Konfigurationsdatei, ACL-Abschnitt

Übersicht



Die Konfigurationsdatei, ACL-Abschnitt

Erstes Beispiel

```
begin acl
```

```
acl_check_rcpt:
```

```
    accept domains = +local_domains
```

```
    accept hosts   = +relay_from_hosts
```

Die Konfigurationsdatei, ACL-Abschnitt

ACL-Verben (Befehle)

■ accept

- Akzeptiert die mail

■ deny

- Die Mail wird abgelehnt

■ drop

- Die Mail wird ohne Benachrichtigung abgelehnt

Die Konfigurationsdatei, ACL-Abschnitt

ACL-Bedingungen

- **hosts**
 - Name/IP des einliefernden Rechners
- **domains**
 - Domainname des Ziels
- **local_parts**
 - Überprüft den Namen vor der Domain (@)
- **verify**
 - Prüft die "Existenz" des Recipienten/Senders
 - verify = sender
 - verify = recipient

Die Konfigurationsdatei, ACL-Abschnitt

ACL-Beispiele

beispiel_acl:

```
accept domains    = +local_domains  
    local_parts = postmaster
```

```
accept domains = +local_domains  
    verify = sender  
    verify = recipient
```

```
deny domains = !+local_domains
```


Die Konfigurationsdatei, ACL-Abschnitt

ACL-Option: message

■ message

- Definiert eine Nachricht (deny)

test_acl:

accept domains = +localdomains

deny message = Keine Mail fuer mich -- Pech
domains = !+local_domains

Die Konfigurationsdatei, ACL-Abschnitt

ACL-Option: endpass

■ endpass

- Kann einen deny erzwingen (accept)

```
accept domains    = +relay_to_domains
    verify        = recipient
```

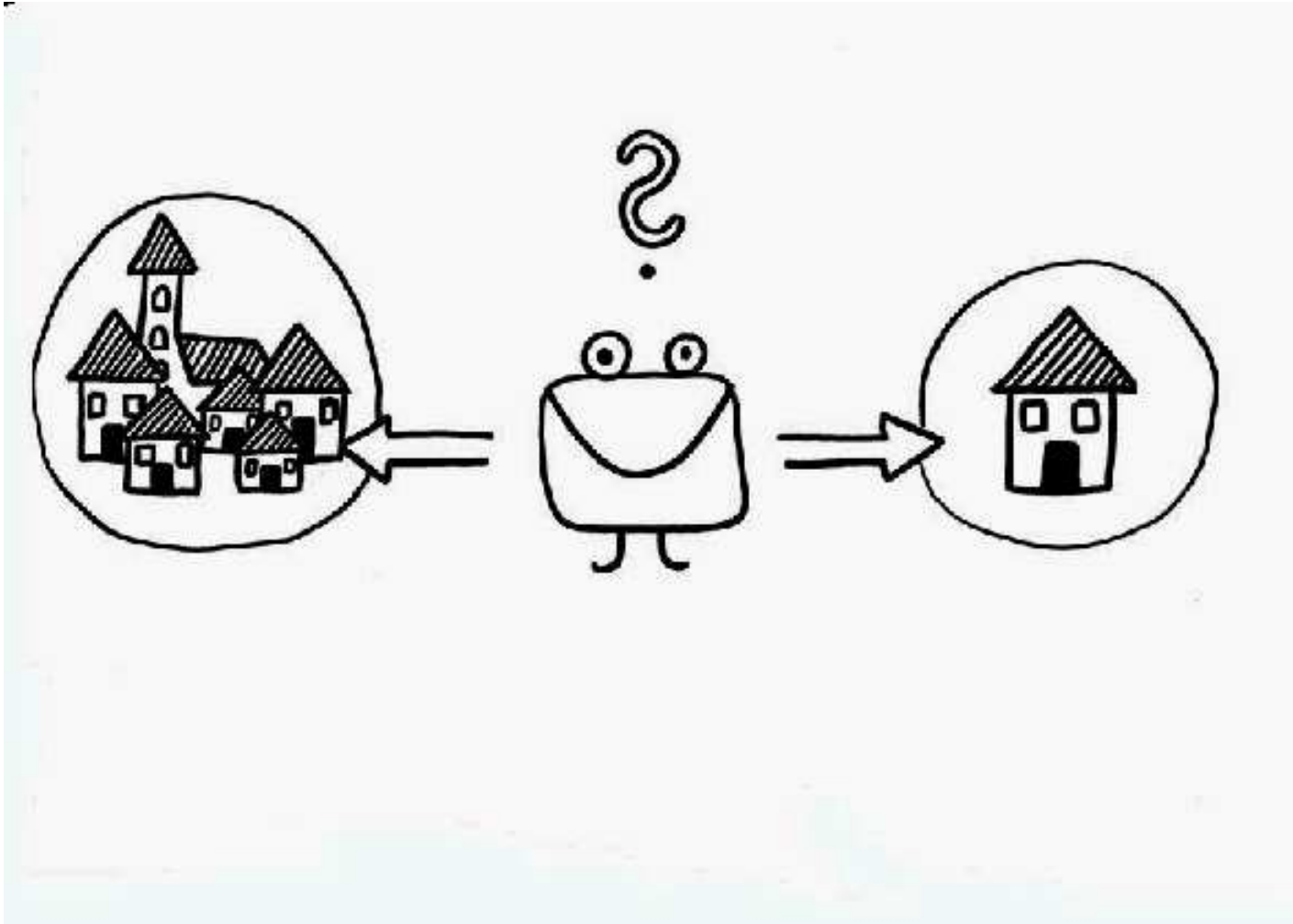
```
deny  message     = recipient not verified
    domains       = +relay_to_domains
```

mit endpass:

```
accept domains    = +relay_to_domains
    endpass
    message       = recipient not verified
    verify        = recipient
```

Router

oder wo lang mit der Mail



Router Übersicht

begin router

Router1:

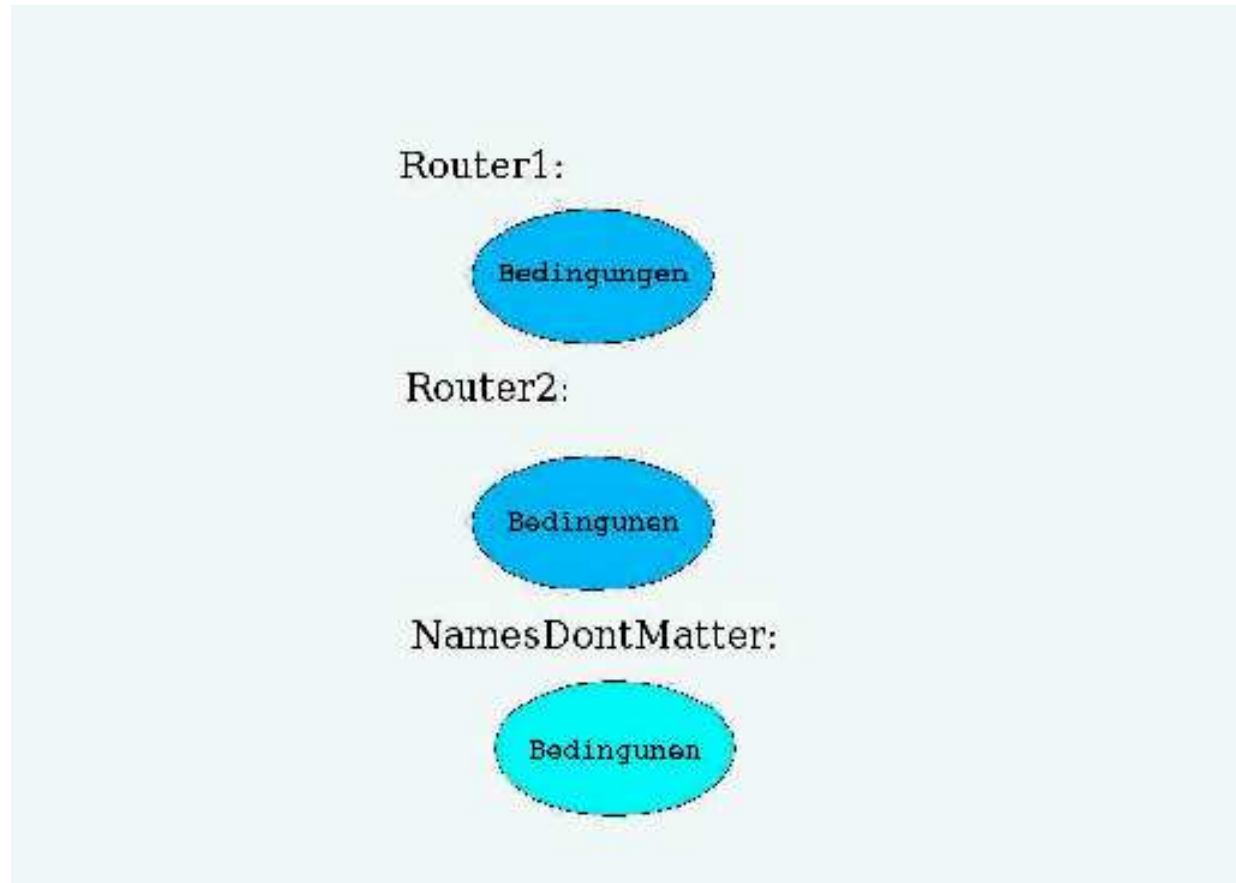
Router2:

NamesDontMatter:

Router werden nacheinander durchlaufen

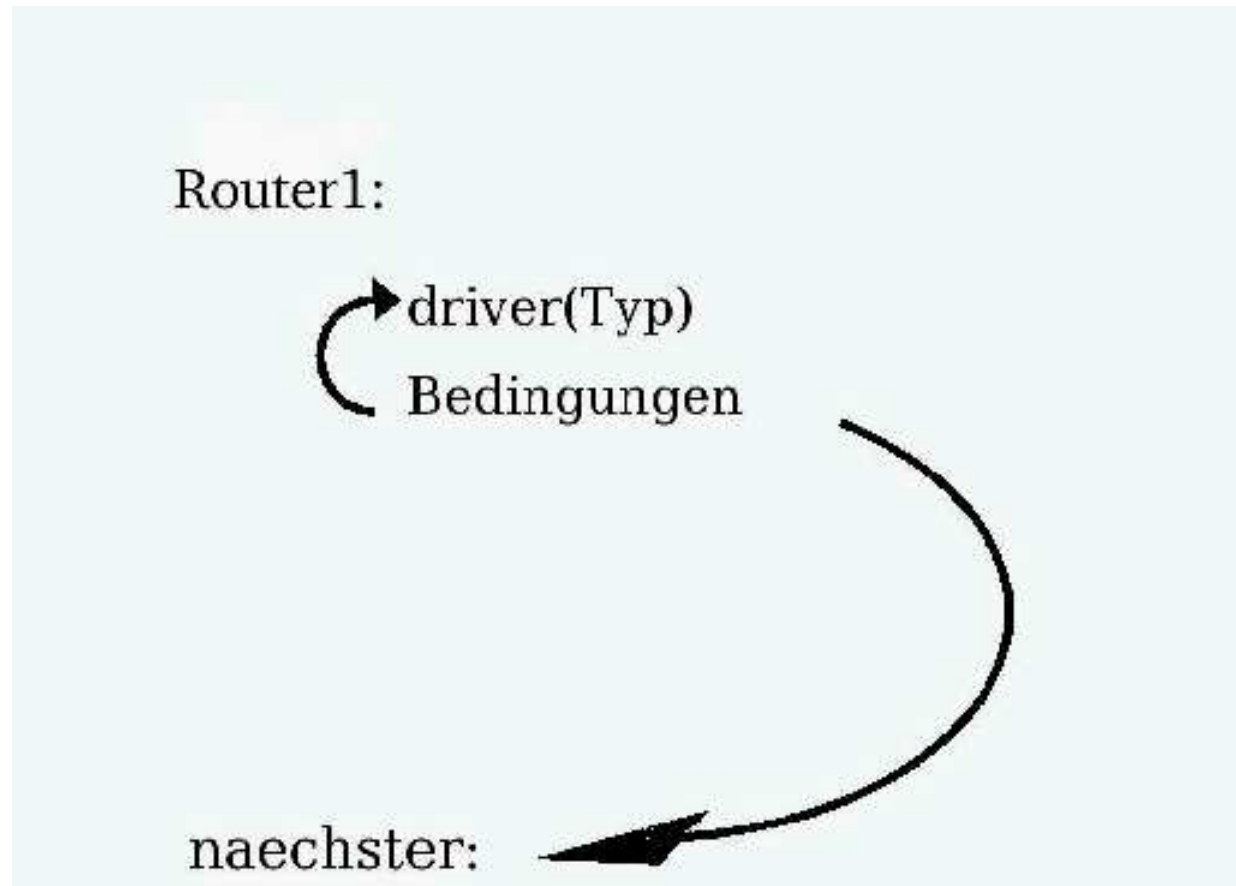
Router Übersicht

begin router



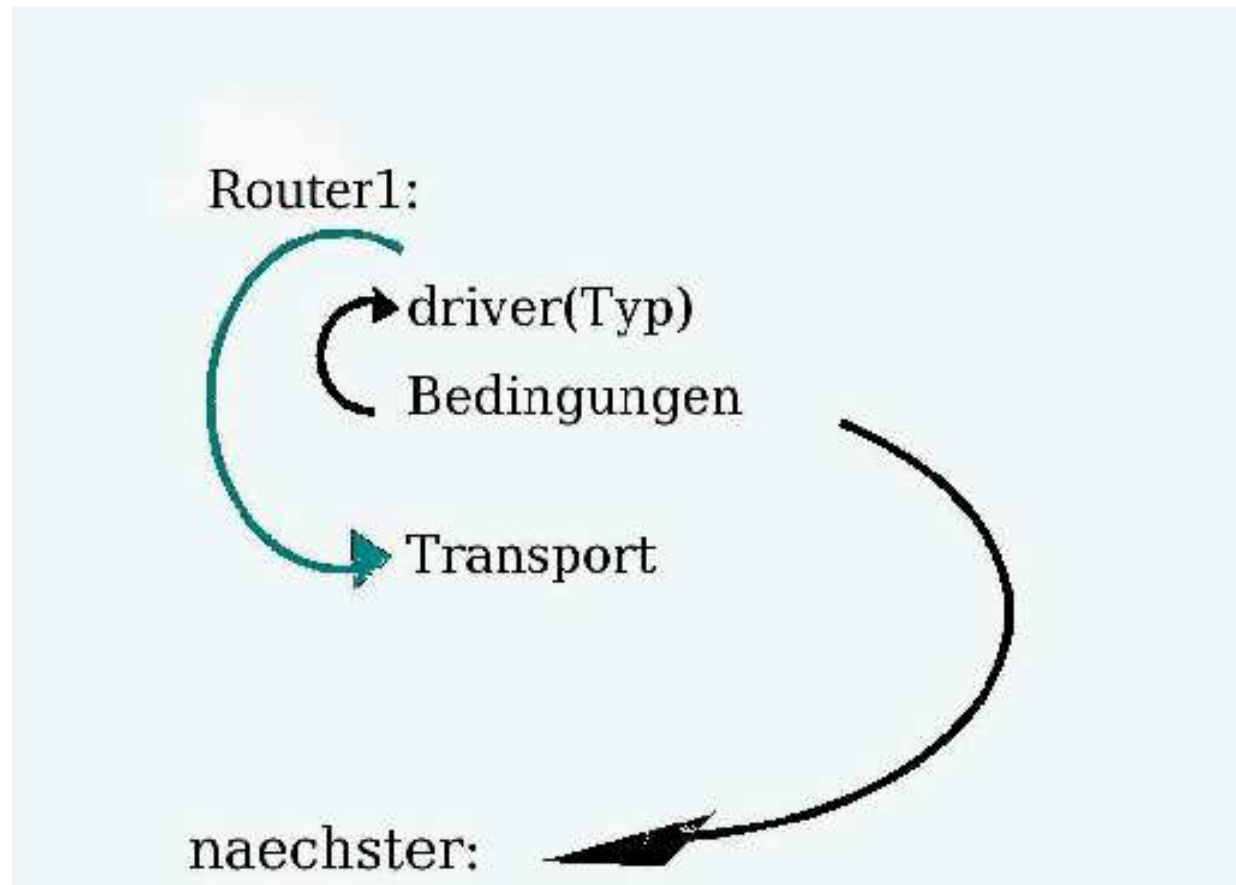
Sind die Bedingungen des Routers erfüllt, wird der Router "benutzt"

Router Übersicht



Bedingung erfüllt -> Router verarbeitet Mail wie bei driver angegeben.

Router Übersicht



War dies erfolgreich, übernimmt der angegebene Transport den Transport der Mail

Router

dnslookup:

driver = dnslookup

domains = ! +local_domains

transport = remote_smtp

- Routerabschnitt beginnt mit
 - begin router
- Router beginnen mit
 - ROUTERNAME:
- driver setzen (Zielerkennung)
- transport bestimmt den Transport
- Exim durchläuft Router in der Konfigdatei
- Sind Bedingungen erfüllt, wird Router ausgeführt
- Passt kein Router, wird die Mail bounced

Router Typen(driver)

■ driver = dnslookup

- findet Zieladresse mittels DNS

■ driver = manualroute

- Zieladresse selbst setzen
- route_list

Option setzt die Route

z.B.: route_list = * smtp.server.de

route_list = *.foo.bar bar.foo

■ driver = accept

- Akzeptiert die Mails
- Sucht keine Ziele

Router Bedingungen

■ hosts

■ local_parts

■ domains

Router transport

transport = via_smtp_verschicken

- transport bestimmt Transporter

Router Beispiele

begin router

wegschicken:

driver = dnslookup

domains = ! +relay_to_domains

transport = remote_smtp

localuser:

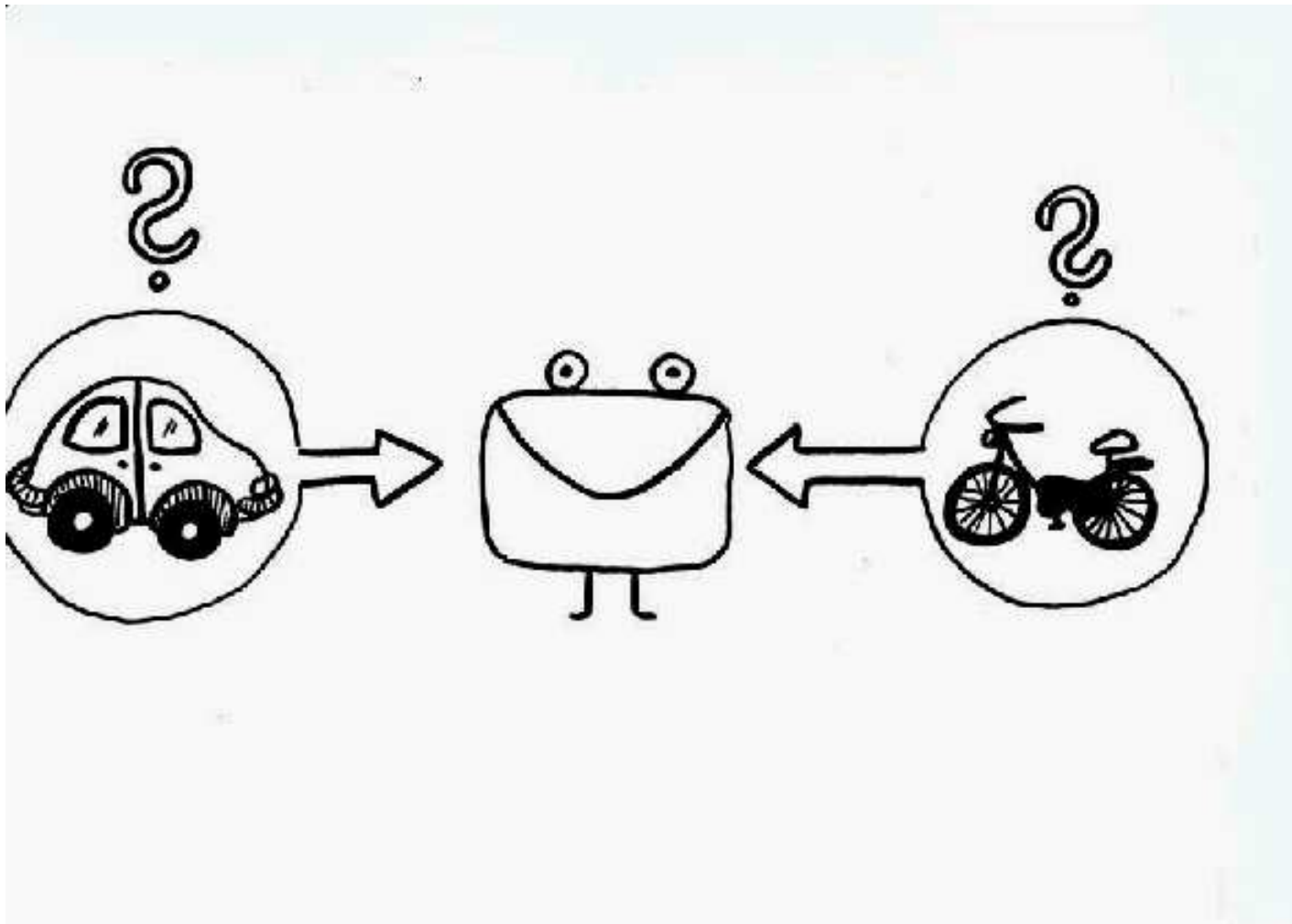
driver = accept

domains = +local_domains

transport = local_delivery

Transports

Wer nimmt mich mit?



Transports

- Weiterer Abschnitt der Konfigdatei

begin transports

- TRANSPORTSNAME:

- driver(Transportart)

Transports

Typen(smtp-driver)

■ smtp

- Verschicken via SMTP
- driver = smtp

Optionen

■ hosts

- Bestimmt Zielrechner

■ hosts_override

- hosts überschreibt Router-Vorgaben

Beispiel smtp

begin transports

remote_smtp:
 driver = smtp

smarthost:
 driver = smtp
 hosts = smart.mailer.de
 hosts_override

Transport Typen(appendfile-driver)

■ appendfile

- Mail in Datei speichern
- driver = appendfile

Optionen

■ user

- uid

■ group

- gid

■ file

- Zielfile zum speichern der Mail

Beispiele

begin transports

local_delivery:

driver = appendfile

file = /var/mail/\$local_part

group = mail

Beispiele

Smarthost,gmx

begin router

```
smarthost:  
driver      = manualroute  
domains     = !+local_domains  
route_list  = * mail.gmx.de  
transport   = gmx_transport2
```

begin transport

```
gmx_transport2:  
driver       = smtp  
headers_rewrite = *@* "mein kreuz <jehova@gmx.de>" fsrw
```

Beispiele

Smarthost,gmx

begin router

```
smarthost:  
driver    = accept  
domains   = !+local_domains  
transport = gmx_transport
```

begin transport

```
gmx_transport:  
driver      = smtp  
hosts       = mail.gmx.de  
headers_rewrite = *@* "mein kreuz <jehova@gmx.de>" fsw
```

tschazu

Fin