

# The Big Picture

## Übersicht rund um E-Mail

Dirk Geschke



Linux User Group Erding

23. Februar 2011

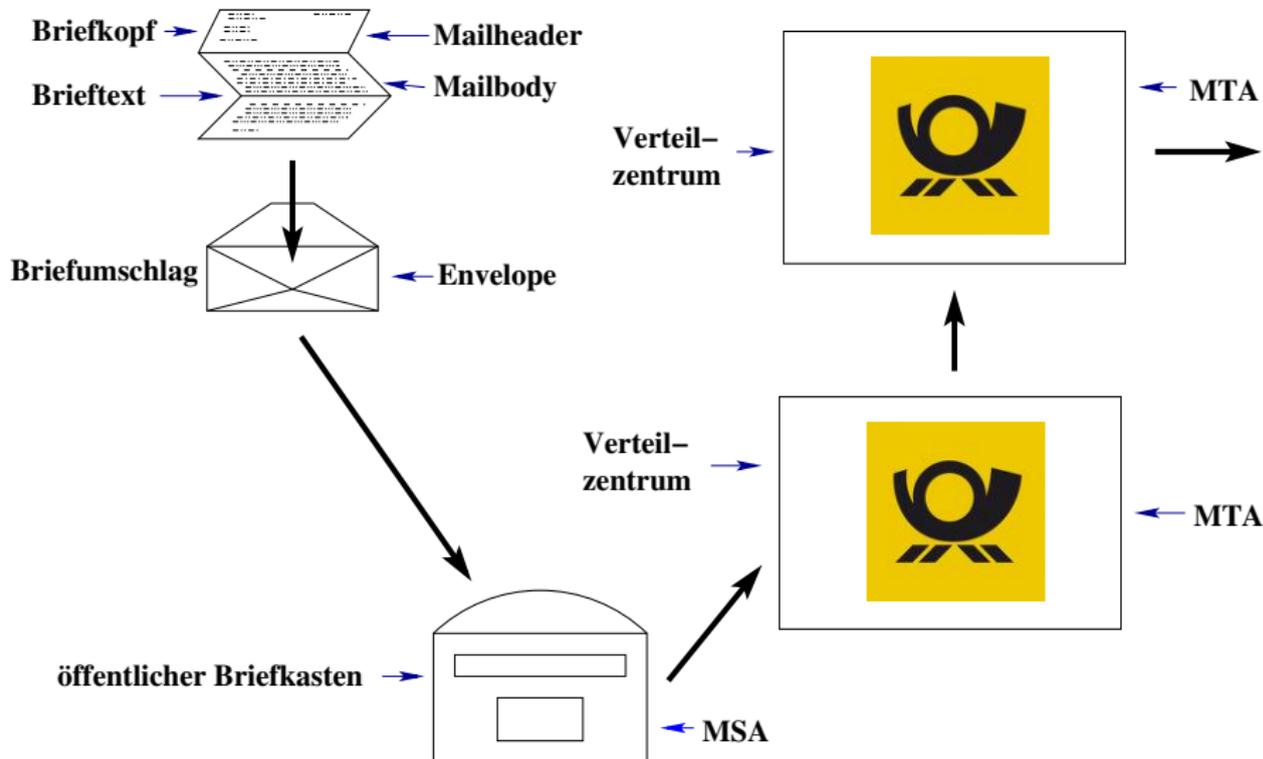
# Gliederung

- 1 Einleitung
- 2 Überblick
- 3 Mailzustellung
- 4 Protokoll
- 5 Mailinglisten und Aliase
- 6 E-Mail-Adressen
- 7 Aufbau von E-Mails
- 8 E-Mail Etikette

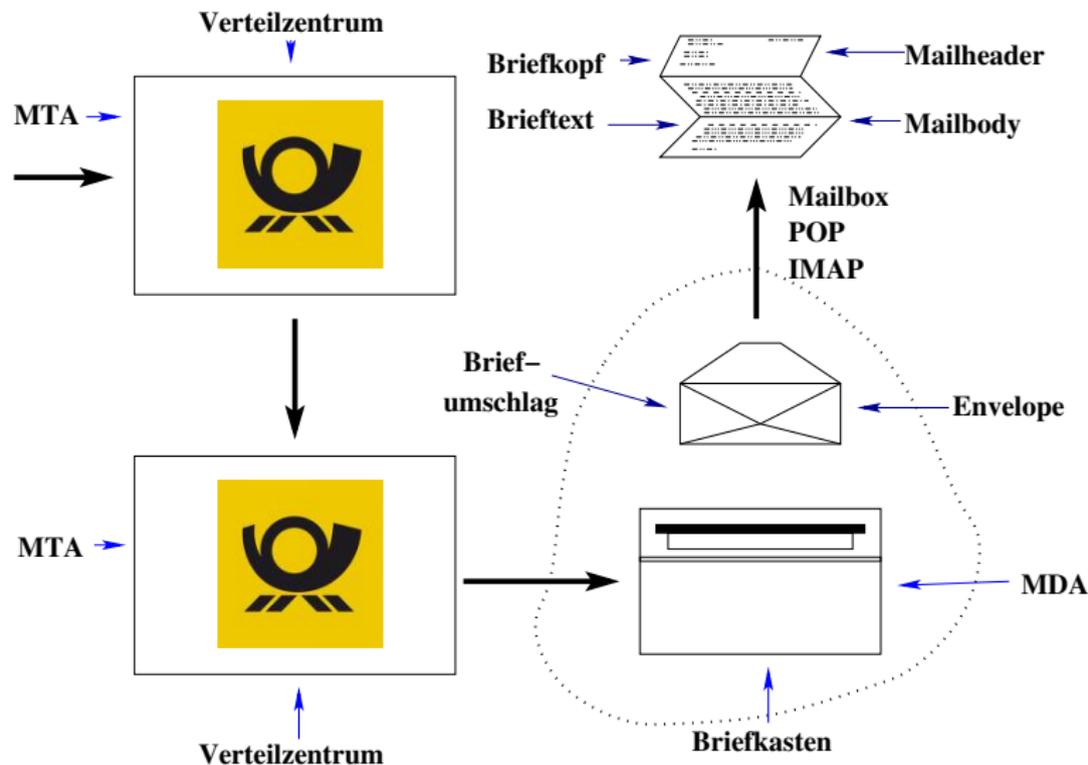
# Der E-Mail Standard

- Noch immer das **wichtigste** Kommunikationsmittel im Internet
- Es gab ursprünglich viele **inkompatible** Mailsysteme, viele basierten auf Erweiterungen von FTP.
- Standard für das Übertragungsprotokoll:  
**RFC-821: Simple Mail Transfer Protocol.**
- Ein einheitlicher Standard für den Aufbau von E-Mails:  
**RFC-822: Standard for the Format of ARPA Internet Text Messages.**
- Aktuelle Versionen sind **RFC-5321** und **RFC-5322**

# The Big Picture - Teil 1



# The Big Picture - Teil 2



# Begriffe

**MUA** Mail User Agent: Mailprogramme/Mailclients wie z.B. Thunderbird, Evolution, Mutt, . . .

**MSA** Mail Submission Agent: der erste Mailserver der verwendet wird.

**MTA** Mail Transfer Agent: ein Mailserver auf dem Weg zum Ziel

**MDA** Mail Delivery Agent: der zustellende Mailserver

**Mailgateway** Mailserver der E-Mails in ein **anderes** System zustellt, z.B. X400

**Relaying** Annehmen **und** Weiterleiten von E-Mails

# Bestandteile einer E-Mail

Eine E-Mail besteht aus 3 Teilen:

- **Envelope**: gehört zum Transportprotokoll
- **Header**: Der Briefkopf einer E-Mail
- **Body**: Der eigentliche Inhalt der E-Mail
- Der Envelope ist normalerweise nicht Bestandteil des Headers bzw. der E-Mail, d.h. der Empfänger bekommt diesen **nicht** zu sehen.

# Auf dem Weg zum Ziel

- Für die Übertragung wird ein eigenes Protokoll verwendet:  
**E**xtended **S**imple **M**ail **T**ransport **P**rotocol
- E-Mails können über mehrere Mailserver **relayed** werden.
- das Protokoll basiert auf **TCP** und verwendet Port **25**
- Die Server werden via DNS ermittelt: **MX**-Records, z.B.:  
lug-erding.de. IN **MX** 100 mail.lug-erding.de.
- bei mehreren MX-Einträgen: der mit der **niedrigsten** Priorität bekommt den ersten Zustellversuch
- existiert kein MX-Record so wird der **A**-Record verwendet.
- jeder SMTP-Server auf dem *Weg* einer E-Mail wird als **hop** bezeichnet

# Allgemeines zum Protokoll

- ein **Mailserver** trägt die Verantwortung für angenommene E-Mails, d.h. entweder wird eine E-Mail korrekt **zugestellt** oder es wird eine **Fehler**-E-Mail generiert.
- ein Mailserver antwortet auf jede Eingabe des Clients mit einem **OK**, einer Aufforderung **weitere Daten** zu senden, einem **temporären** oder **permanenten** Fehler.
- Ausnahme: **PIPELINING**, wenn vom Server unterstützt
- Die Kommunikation ist in der Regel **zeilenorientiert**, Abschluss ist immer **CRLF** (#0D#0A oder unix-like \r\n).
- Server antwortet immer mit einem **dreistelligen Zahlencode**, einem Leer- oder Minuszeichen und einem **ASCII-Text**

# Allgemeines

Allgemeines über das Protokoll:

- **Text-Zeilenlänge** ist auf 1000 Zeichen *inklusive* CRLF begrenzt, **Kommandozeilen** auf 512 Zeichen.
- Client-**Kommandos** sind *case-insensitive* und bestehen aus reinem **ASCII**.
- Empfehlung: **Kommandos** *groß* schreiben, einige Server verlangen dies entgegen RFC-821.
- SMTP erlaubt generell nur **7-Bit** Daten, 8. Bit darf gelöscht oder die Annahme verweigert werden.
- **Extended SMTP** ermöglicht auch 8-Bit für den Mailbody
- 8-Bit Zeichen im **Mailheader** sind nur via **MIME** in 7-Bit Darstellung möglich.

# Kommunikation mit einem Mailserver

- 1 **Client** öffnet TCP-Verbindung zu Port **25** des Servers
- 2 **Server** antwortet mit Eröffnungsmitteilung, → kann **Software** und **Version** zur Fehlersuche enthalten, z.B.

```
220 mail.lug-erding.de ESMTP \  
Sendmail 8.13.8/8.13.8/Debian-3; \  
Tue, 17 Mar 2009 21:38:14 +0100; \  
(No UCE/UBE) logging access from: \  
[91.64.98.12] (FAIL)-[91.64.98.12]
```

- 3 **Client** sendet **EHLO**-Kommando mit **FQDN** des Clients, z.B.:

```
EHLO geschke.linuxhome.org
```

Falls der Server oder Client dies nicht unterstützt wird **HELO** verwendet (RFC-821)

# Kommunikation mit einem Mailserver

- 4 Server antwortet mit **unterstützen** Kommandos / Erweiterungen (bei **EHLO**), z.B.:

```
250-mail.lug-erding.de Hello [92.64.98.12], please send  
250-ENHANCEDSTATUSCODES  
250-PIPELINING  
250-8BITMIME  
250-SIZE  
250-DSN  
250-AUTH DIGEST-MD5 CRAM-MD5  
250-DELIVERBY  
250 HELP
```

**Minus**-Zeichen an der 4. Stelle: es gibt weitere Zeilen!

# Kommunikation mit einem Mailserver

- 5 **MAIL FROM:** *<reverse-path>* [*<mail-parameter>*]

Der *reverse-path* enthält idR den Absender, Fehler-E-Mails werden dorthin gesendet z.B.:

```
MAIL FROM: <Dirk@geschke-online.de>
```

- 6 Wird der Sender akzeptiert so sendet der Server **250 OK**

- 7 **RCPT TO:** *<forward-path>* [*<rcpt-parameter>*]

Der *forward-path* enthält die Empfängeradresse, z.B.:

```
RCPT TO: <Dirk@lug-erding.de>
```

- 8 Wird der Empfänger akzeptiert so sendet der Server **250 OK** oder **550 no such user**, z.B.:

```
250 2.1.5 <Dirk@lug-erding.de>... Recipient ok
```

# Kommunikation mit einem Mailserver

- 9 Client sendet **DATA**
- 10 Server antwortet idealerweise mit **354 go ahead**, z.B:  
354 Enter mail, end with "." on a line by itself
- 11 Client überträgt Daten **zeilenweise**, Mailheader und Mailbody sind durch Leerzeile getrennt. Den Abschluss bildet die Sequenz **"CRLF . CRLF"**, z.B.:

```
Subject: Just a Test  
From: <Dirk@geschke-online.de>
```

```
nix besonderes
```

```
.
```

# Kommunikation mit einem Mailserver

- 12 Server sendet **250 OK**, z.B.:

```
250 2.0.0 15L81DFD032735 Message accepted \
for delivery
```

- 13 Beenden der Übertragung, Client sendet **QUIT**

- 14 Server sendet **221 OK** und schließt die Verbindung:

```
221 2.0.0 mail.lug-erding.de closing connection
```

Das ist im wesentlichen alles, es können aber auch **Fehlercodes** ausgegeben werden und es gibt noch Kommandos für die Fehlersuche sowie ein paar **Erweiterungen**.

# Numerische Antwort-Codes des Servers

Die 1. Stelle besagt:

- 1yz Kommando akzeptiert aber noch nicht ausgeführt
- 2yz Kommando wurde erfolgreich ausgeführt
- 3yz positive Zwischenmeldung, weitere Daten erforderlich
- 4yz temporärer Fehler
- 5yz permanenter Fehler

# Numerische Antwort-Codes des Servers

Die 2. Stelle besagt:

- x0z Syntax-Fehler: nicht implementiert, überflüssig, falsche Reihenfolge
- x1z Information: Antwort auf Informationsfrage wie Status oder Hilfe
- x2z Verbindung: Alles rund um die Verbindung
- x3z un spezifiziert
- x4z un spezifiziert
- x5z Mailsystem: Status des Empfangssystems bzgl. Mailtransfer oder anderer Mailaktionen

Die 3. Stelle gibt eine feinere Bedeutung pro Kategorie an.

# Relayverarbeitung angenommener E-Mails

- DSN, **D**elivery **S**tatus **N**otifications, *Postmaster* E–Mails: Benachrichtigung über den Status einer E-Mail, formalisiert
- Benachrichtigung an Absender aus dem Protokoll  
→ **E**nvelope–Adresse
- Absender ist die **N**ULL–Adresse: <>
- Mailserver muss eine **R**eceived–Zeile im Mailheader voranstellen:
  - ▶ FROM–Feld: Adresse des sendenden Mailservers (aus EHLO–Zeile und IP)
  - ▶ ID–Feld: ID der E–Mail in der Logdatei
  - ▶ BY–Feld: optional, von wem (System, Software) wurde die E-Mail empfangen
  - ▶ FOR–Feld: optional, die Liste der Empfänger
  - ▶ WITH–Feld: Methode
  - ▶ Datums–Feld: lokale Zeit mit Offset
- Letzter Mailserver fügt **r**eturn–**p**ath hinzu: Absender aus Envelope (Fehlerbehandlung z.B. von Mailinglisten)

# Mailinglisten und Aliase

- **SMTP**-Server sollten beides unterstützen
- bei Listen: **MAIL FROM** sollte den **Listenverwalter** angeben (wegen Fehlermails)
  - Alias** Mailserver ersetzt die Zieladresse im **Envelope**, der Rest bleibt unverändert
  - Liste** Zieladresse im Envelope werden durch die Adressen der **Liste** ersetzt, die Absendeadresse im Envelope wird die des **Listenverwalters**

# Format von E-Mail Adressen

- **Mailbox**: *Displayname* <local-part@domain> / local-part@domain
- **Mailbox-Liste**: Komma-separierte Liste: mailbox1, mailbox2
- **Mailgruppe**: *Gruppenname*: mailbox1, mailbox2, ... ;
- **local-part**: *atext* [ "." *atext* ] / *quoted-string*
- **atext** kann bestehen aus  
**a-z, A-Z, 0-9, !, #, \$, %, &, \*, +, -, /, =, ?, ^, \_, ' , { , |, }, ~**
- **quoted-string**: **NO-WS-CTRL**, *Rest* von **ASCII** außer Backslash und "
- Im **Envelope** ist kein Displayname erlaubt, die Zeichen < und > sind Pflicht!

# E-Mailheader

- grundsätzlich nur **7**-Bit Zeichen. Umwandlung von 8-Bit Zeichen via **MIME** möglich
- Header Felder bestehen aus **Feldnamen** und **“:”** gefolgt vom **Feldwert**
- es gibt **strukturierte** und **unstrukturierte** Felder
- **RFC-5322** ist für das Format verantwortlich
- **keine** feste Reihenfolge, kann gelegentlich geändert werden, **sollte** aber nicht!
- **Trace**-Felder, **“Received**-Zeilen”, dürfen **nicht** geändert werden!
- gleiches gilt für **Resent**-Felder!
- diese Felder sollen **vorangestellt** werden
- **Pflichtfelder**: **Absende-Datum** und **Absender**

# Headerfelder – eine Auswahl

**Date** Versanddatum

**From** Absender, Displayname optional, z.B:  
From: "Dirk Geschke«kontakt@lug-erding.de>

**To** Empfänger, Displayname optional

**Cc** Carbon-Copy: sichtbare Kopie an angegebene  
Adresse(n)

**Bcc** Blind-Carbon-Copy: unsichtbare Kopie an angegebene  
Adresse(n)

**Reply-To** Antworten an angegebene Adresse(n)

**Subject** Betreff-Zeile

**Message-ID** eindeutige Message-ID

**In-Reply-To** ist Antwort auf Message-ID

**References** dito, kann für Threads verwendet werden

# MIME: Multipurpose Internet Mail Extensions

- relevante **RFCs**: RFC-2045 bis RFC-2049 und RFC-4288/4289
- SMTP hat zwei wesentliche **Probleme**:
  - 1 Zeilenlänge ist auf **1000** Zeichen begrenzt
  - 2 **7-Bit** ASCII
- Lösung besteht in der Verwendung von **MIME**
- **5** neue Felder für den **Mailheader**
- Möglichkeit der **Kodierung** von Daten beim Transfer
- **Identifikation** des Datentyps zwecks Darstellung im Client

# Content-Transfer-Encoding

**7bit** Daten sind reine 7-Bit ASCII, kein Encoding, maximale Zeilenlänge: 1000 Zeichen

**8bit** Daten sind reine 8-Bit, kein Encoding, maximale Zeilenlänge: 1000 Zeichen

**binary** Daten sind rein binär, d.h. keine Zeilenstruktur aber auch kein Encoding

**quoted-printable** größtenteils lesbar, nur **nicht-7**-Bit Daten werden konvertiert

**base64** Daten sind im Base64-Format, nicht lesbar

**IANA-** oder **x-token** Format muss anderweitig spezifiziert sein

# Quoted-Printable

- anzuwenden wenn das meiste **ASCII** ist
- **8-Bit Darstellung**: “=”-Zeichen plus 2-stelligen Hexcode in **Großbuchstaben**, z.B. “=” → “=3D”
- **ASCII**-Zeichen 33-60 und 62-126 können normal verwendet werden
- Zeilenlänge maximal **76** Zeichen, Verwendung von = am Ende einer Zeile als **Soft Line Break**
- **Empfehlung**: ! " # \$ % & ' ( ) \* + , - . : ; < = > ? [ \ ] ^ \_ { | ~ } ebenfalls **umwandeln**

# Base64

- anzuwenden bei hauptsächlich **binären** Daten
- Darstellung durch **65** ASCII-Zeichen
- **6-Bit** Darstellung in 24-Bit Gruppen, 4-Base64 Zeichen ergeben 3 Byte
- “=”-Zeichen als **Padding**
- **ASCII**-Zeichen **a-z**, **A-Z**, **0-9**, **+** und **/**
- Zeilenlänge maximal **76** Zeichen

- Zwei Varianten:
  - 1 Simpler Anhang mit weiteren Informationen über den E-Mail-Versender, teilweise bei gewerblicher Nutzung Pflicht (vergleiche Impressum)
  - 2 Digitale Unterschrift zur Verifizierung des Autors: Prüfsumme über den Inhalt der E-Mail (Mailbody) die mit private-Key verschlüsselt angehängt wird

# Etikette für E-Mails

- E-Mails sind zwar formlos aber sollten eine **Form** haben
- **From**: wenn nicht eindeutig Displaynamen verwenden
- **Subject**: kurzer aussagekräftiger Text
- **Re: AW: Re: FWD:**–Ketten vermeiden
- **Abwesenheitsbenachrichtigungen** vermeiden
- keine **HTML** E-Mails
- **Anrede** und abschließende **Grußformel** verwenden
- **Signatur** mit näheren Informationen verwenden: Wichtig  
Grußformel **nicht** in die Signatur setzen!

# Etikette für E-Mails

- **Text** deutlich unter 80 Zeichen pro Zeile, gelegentlich Absätze einfügen
- bei **Antworten** nur die Teile zitieren auf die man sich bezieht
- bei Gefühlen wie ironische Anspielungen **Emoticons** verwenden ;-)
- **Disclaimer**: weglassen! Sie haben ohnehin keine rechtliche Relevanz.
- möglichst eine **Rechtschreibung** befolgen
- bei **Problemen**: möglichst präzise Beschreibung was nicht geht
- bei Mailinglisten das Subject **taggen**, z.B. [LUG-Erding] hinzufügen
- Details sind hier zu finden

<http://www.lug-erding.de/kontakt/netikette.html>

# Geschafft!