



Ein Mailsystem mit Vorverarbeitung und Datenbankspeicherung

Tilo Buschmann

15. April 2003



Fullscreen

Quit



In diesem Vortrag werde ich eingehen auf

- Wünsche und Anforderungen von Webmailanbietern
- Die Vorverarbeitung von E-Mails zur Geschwindigkeitsgewinnung
- Das Halten einer Vielzahl an Informationen über den momentanen Zustand aller E-Mails in einer Datenbank für eine größere Flexibilität

Gliederung:

- Hinleitung zum Konzept
- Konzept
- Proof-of-Concept-Implementation



Fullscreen

Quit



Hintergrund:

- Nutzer sind nur sehr schwach an Internetserviceprovider gebunden
- kostenlose Mailanbieter füllen eine Lücke
- Mailanbieter wollen Geld verdienen
- Der Funktionsumfang und Komfort muss konkurrenzfähig sein

Gewollter Funktionsumfang

- Zugang über Interfaces für die EMailabfrage, den Emailversand und eine generelle Verwaltung der eigenen Einstellungen
- Zugang über einen normalen EMailklienten auf den selben EMailumfang
- ein riesiger Mehrwert der sich nahtlos in den Rest integriert



Fullscreen

Quit



Der Funktionsumfang des Freemailanbieters web.de umfasst (nur relevante Dienste):

- Emailempfang und Versand, Begrenzung auf 4MByte
- Webzugang, IMAP4- und POP3-Zugang, SMTP, auch verschlüsselt
- SMS
- Faxe empfangen und verschicken
- Grußkarten
- Termine

Zum Geldverdienen gehört:

- Mehr Speicher
- Mehr SMS
- Weckrufe
- ...



Fullscreen

Quit



Warum sehe ich hier nun Probleme?

An welchen Stellen des Funktionsumfangs kann man noch optimieren oder muss über das übliche Hinaus gehen?

Wie entstehen Konflikte zwischen den drei Punkten des Grundumfangs von Mailanbietern?

Natürlich müssen nicht alle folgende Wünsche und Vorstellungen aus wissenschaftlich-technischer Sicht vernünftig sein ...



Fullscreen

Quit



Problemklasse der zeitnahen Konfiguration

- es müssen mehrere 100.000 Nutzer auf verschiedenen Servern verwaltet werden
- das ganze in einem möglichst virtuellen Kontext
- Konfigurationswünsche sollen flexibel sein und sich sofort umsetzen lassen
- Einfaches Pop after SMTP und natürlich SMTP Auth



Fullscreen

Quit



Problemklasse Webinterface vs. Imap

- Das Webinterface könnte selbst Imap benutzen, um auf den selben Datenbestand wie externe Programme zuzugreifen
- Dabei stößt man auf Beschränkungen des Protokolles
- Wenn Emails immer in Reinform vorliegen, müssen diese beim Auswählen u.U. ärgerlich langsam decodiert werden.
- Beispiel: Schwierige Webinterfacefunktion^a
- Beispiel: Attachments

^a<http://www.ietf.org/proceedings/00jul/I-D/imapext-regex-00.txt>



Fullscreen

Quit



Problemklasse Administration und Management

Die Firma sowie die Administration will gerne auf zusätzliche Informationen und Informationsabrufmöglichkeiten zugreifen.
Dazu zählt z.B. eine Verbindung vom Emailinformationen mit Kundeninformationen.



Fullscreen

Quit



Es gibt natürlich vielfältige Lösungsmöglichkeiten abhängig von den eigenen Bedürfnissen, Vorzügen und Geldmitteln.

- IMSP^a
- Hotmail mit einem eigenen Protokoll
- cachendes Webinterface
- etwas ganz anderes

^a<http://asg.web.cmu.edu/cyrus/rfc/imsp.html>



Fullscreen

Quit



Meine hier vorgesehene Antwort geht einen eigenen Weg.
Dabei kommen zur Anwendung:

- Die Konfiguration der SMTP-, IMAP4- und POP3-Server liegen in einer Datenbank vor
- Der MTA verarbeitet die Email vor dem Ablegen im Dateisystem
- Eine Synchronisation zwischen den Emails im Dateisystem und einer Datenbank
- sämtliche zusätzlich benötigten Informationen (alle für den Betrieb der Services) liegen selbstverständlich auch in der Datenbank vor

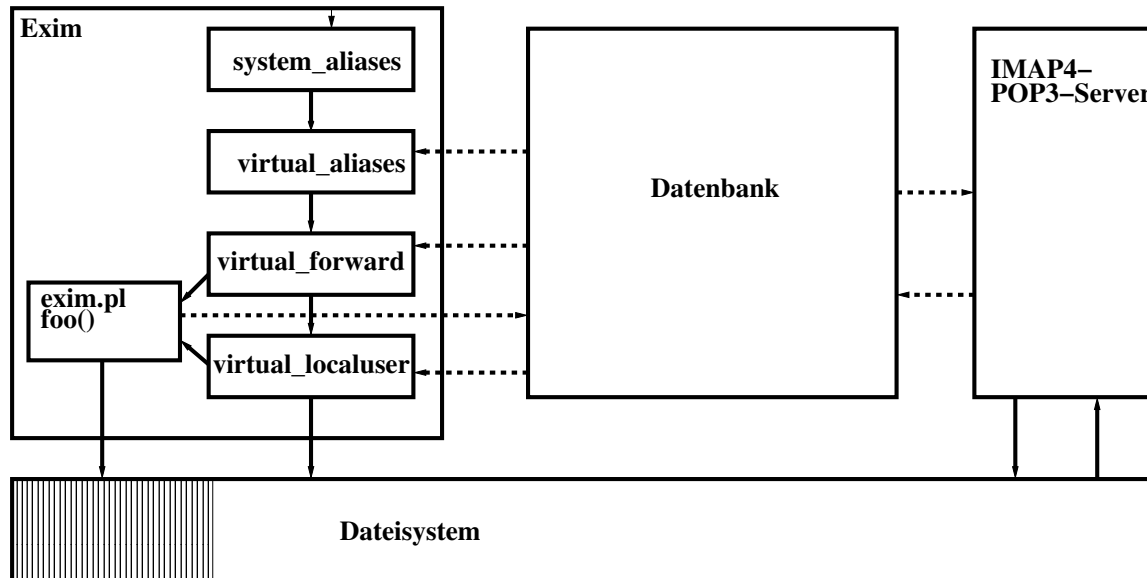


Fullscreen

Quit



Weg einer Email



Legende:

————> Bearbeitungsweg einer Email in Reinform

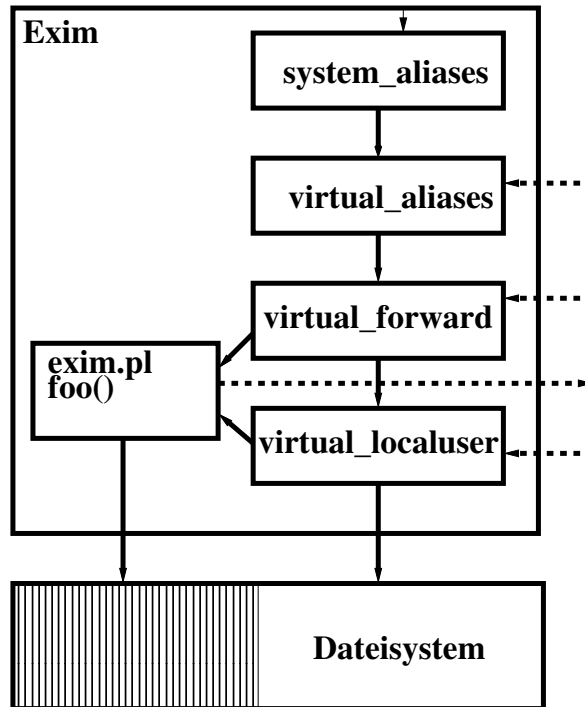
- - - - -> Austausch mit Datenbankinformationen



Fullscreen

Quit

Der Eingang ins System



Legende:

————> Bearbeitungsweg einer Email in Reinform

- - - -> Austausch mit Datenbankinformationen

- Gebraucht wird ein MTA mit DB-Anbindung, im besten Fall Skriptingfähigkeit, Run-Time-Konfiguration, SMTP-Auth und SSL.
- Ein traditioneller Abgleich mit einer Systemalias-Datei
- Ein Abgleich mit einer virtuellen Aliassammlung, traditionelle virtusertable und aliasfile zugleich
- Ein Durchlauf durch Filter
- Die Email wird im Maildirformat abgelegt und zusätzlich direkt einem Skript übermittelt



- Das Skript übernimmt das Parsen der Email
- Attachments, Header und Body werden getrennt gespeichert
- Der Header wird ausgewertet, die Email wird inklusive eindeutiger Message-ID in die DB gespeichert
- In den Maildir-Emails und der Datenbank werden die selben Message-IDs genutzt
- Beim Parsen der Original-Email wird eine eigene ID generiert, die extra in der Datenbank gespeichert wird
- Dieses Skript lässt sich kombinieren mit anderen Services, bei denen die Email zerlegt werden muss, z.B. innerhalb eines Virenfilters
- Nach dem Durchlauf dieses Skriptes befindet sich das System im angestrebten Zustand: Die Email ist sowohl in der Maildir-Form für den IMAP4/POP3 Server vorhanden, in Einzelteilen zerlegt für das Webinterface und andere Belange, wichtige Headereinträge befinden sich in der Datenbank



Fullscreen

Quit



- Im günstigsten Fall kommen POP3 und IMAP4 vom selben Server bzw. der selben Serverfamilie
- Der Server muss sämtliche Änderungen in der DB eintragen
- Das erfordert natürlich eine intensive Umprogrammierung
- Probleme stellen hier vor allem Fälle wie das Hochladen von Emails sowie das kopieren von EMailS dar
- Teilweise müssen gecachte Daten gelöscht oder kopiert werden

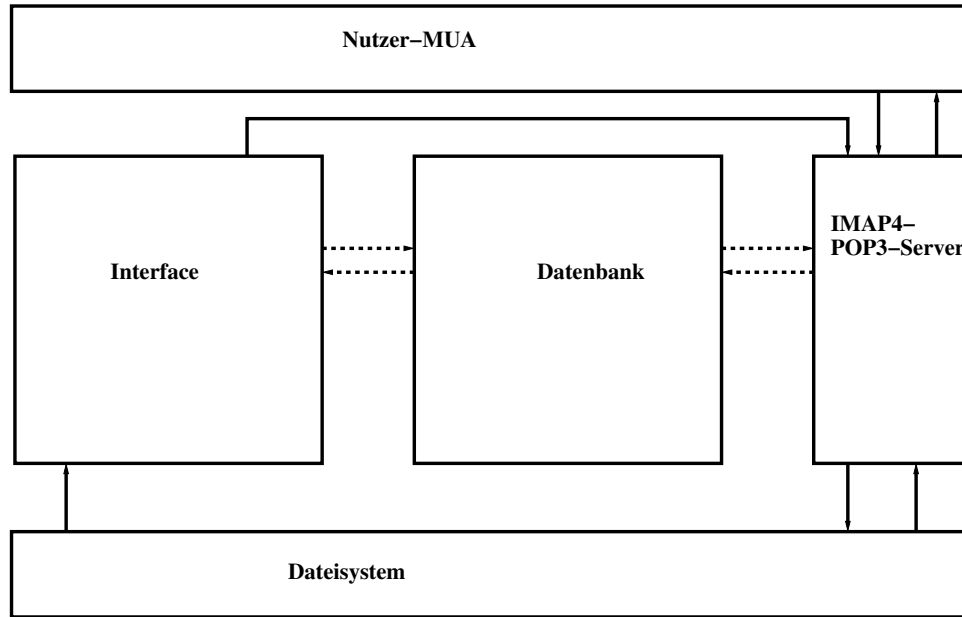


Fullscreen

Quit



Schematischer Zugriff des Webinterface



Legende:

→ Bearbeitungsweg einer Email in Reinform

- - - - -> Austausch mit Datenbankinformationen



Fullscreen

Quit



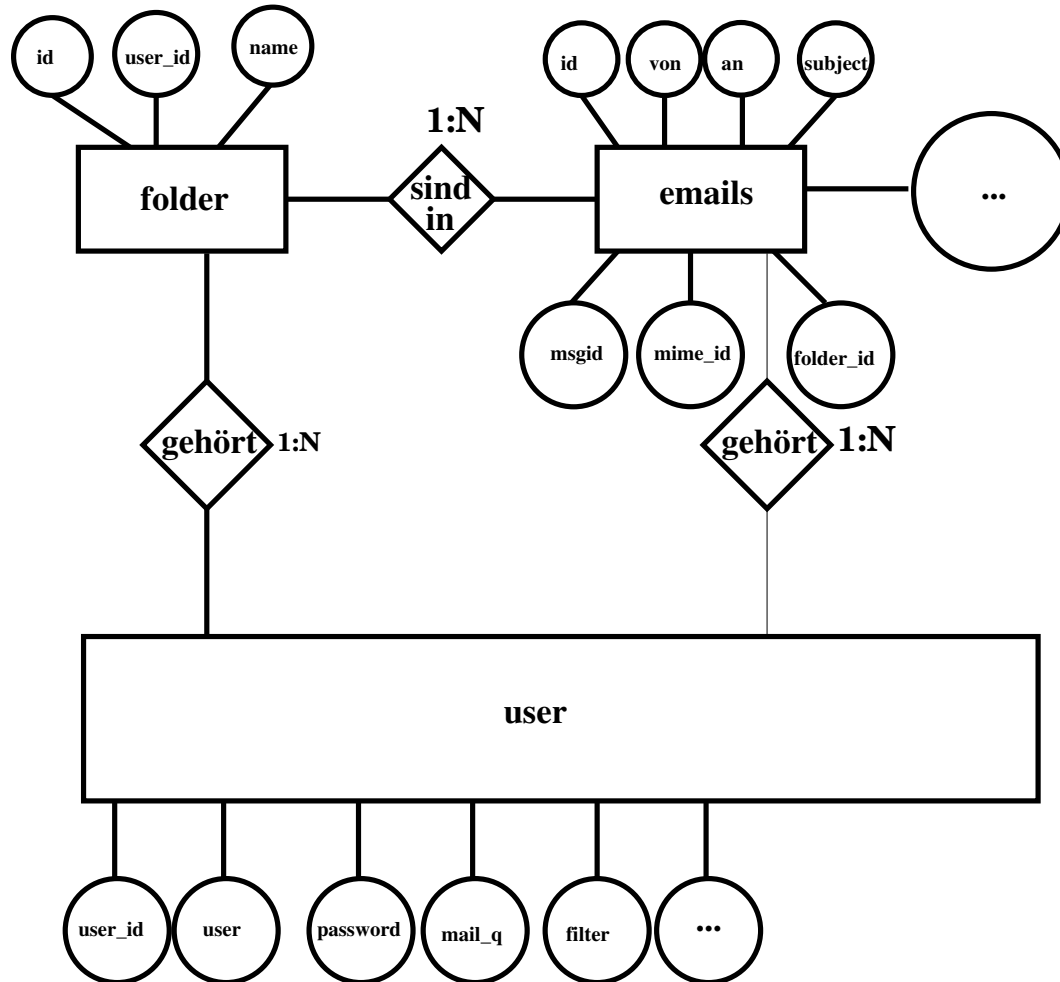
Interfaces Interfaces Interfaces (und Interfaces)

- Verschieden Stufen des Datenzugriffes freier Wahl
- Im Idealfall greift ein Webinterface z.B. immer auf Datenbankinformationen zu
- Nächste Lesestufe sind gecachte Informationen
- Diese Mechanismen sind auch sehr hilfreich für WAP-Interfaces (Handyabhängig) oder Emails, die man sich am Telefon vorlesen lassen kann
- Problem bleibt weiterhin der *schreibende* Zugriff, in diesem Fall sollte man wiederum über das IMAP4 gehen



Fullscreen

Quit



Fullscreen

Quit



Die Implementation: Übersicht

- Proof-of-Concept Implementierung existiert
- Die MTA Wahl ist einfach: Exim
- beherrscht DB-Anbindung, Query-Lookups, String-Expansion, SMTP-Auth, SSL, integriertes Perl, Konfiguration nach Wahl
- als POP3/IMAP4 Server wurde Courier gewählt
- Beherrscht Autorisation bereits per DB
- Lässt sich relativ einfach umprogrammieren, ist simpel zu verstehen
- Beherrscht (nur) Maildir-Format



Fullscreen

Quit



```
MAIN_DOMAIN=crystalship.csn.tu-chemnitz.de
MYSQL_USER = SELECT user FROM user WHERE \
            user like '${quote_mysql:$local_part}' \
            AND valid = 'true'
MYSQL_ALIAS = SELECT ergebnis FROM emailadressen \
            WHERE email like \
            CONCAT('${quote_mysql:$local_part}', \
            '@', '${quote_mysql:$domain}')
USER_TO_USERID = SELECT userid FROM user WHERE \
            user='${quote_mysql:$local_part}'
```



Fullscreen

Quit



```
virtualuserforward:  
  driver = forwardfile  
  domains = MAIN_DOMAIN  
  local_parts = mysql;MYSQL_USER  
  pipe_transport = virtual_address_pipe  
  directory_transport = virtual_address_directory  
  reply_transport = address_reply  
  no_verify  
  file_directory = /var/spool/mail/${lookup\  
    mysql {USER_TO_USERID}{$value}${{local_part}}}  
  check_ancestor  
  check_local_user = false  
  owners = mail  
  user = mail  
  group = mail  
  modemask = 002  
  filter  
  data=${lookup mysql \  
    {SELECT CONCAT('# Exim filter',filter) FROM user\  
  where user = '${local_part}'}}
```



Fullscreen

Quit



```
virtual_delivery:
  driver = appendfile
  maildir_format
  directory = /var/spool/mail/\
    ${lookup mysql {USER_TO_USERID}{$value}${local_part}}
  user = mail
  group = mail
  maildir_tag = "$message_id:2,"
  quota = ${lookup mysql {select\
    CONCAT(mail_q,'M') from user where user = \
    '${quote_mysql:$local_part}'}{$value}}
  mode = 0660
  directory_mode = 0770
  create_directory = true
  check_string = ""
  escape_string = ""
  prefix = ""
  suffix = ${perl{foo}${lookup mysql\
    {USER_TO_USERID}{$value}${local_part}}}${message_id}}
```





- Wesentliche Headervariablen bekommt das Skript von Exim selbst
- Der komplette Text der Email wird aus dem Spool geholt
- Neben den Textteilen, Attachments und “HTML-Nachrichten“ wird ebenfalls der Header und Body getrennt hinterlegt

```
$outputdir = '/var/spool/mailsystem/cached';  
$parser->output_under($outputdir);  
$entity = $parser->parse_data(  
    $message_headers . "\n" . $message_body);  
$results = $parser->results;  
@messages = $results->msgs;
```





- Konfiguration wird über den Authdaemon bereits aus der Datenbank geholt

```
char *querystart = "DELETE_FROM_emails_WHERE_msgid_like_";  
char *queryend = "";  
...  
if (mid != NULL) {  
    query = (char *) malloc(strlen (string1 ) +  
                           strlen (querystart) +  
                           strlen (queryend) + 1);  
  
    strcpy (query, "" );  
    strcat ( strcat ( strcat (query,querystart), string1 ), queryend);  
    mysql_query(&mysql,query);  
    free (query);  
}
```





Notwendige Teilanpassungen:

- POP3: Löschen der zum löschen markierten Emails (cleanup)
- IMAP4:
 - Append
 - Create
 - Delete
 - Rename
 - Store
 - Expunge
 - Copy



Fullscreen

Quit



Kritische Betrachtung und Schlussfolgerung

- Unmodular, muss als Einzellösung an einen bestimmten Fall angepasst werden.
- Starke Eingriffe in bestehende Programme
- Erst- und Folgekosten für Hardware und Programmierung enorm hoch
- Gesamtperformanz nimmt ab, zu Gunsten der Möglichkeiten
- Provider muss eine Abwägung treffen



Fullscreen

Quit