# POP3ye

## Een web-based e-mail client in PHP5

**Author**: Jelle Piekaerts
**Filename**: POP3ye.doc
**Filesize**: 2616320
**Last saved by**: k-rash
**Nr chars**: 40316
**Num pages**: 40
**Num words**: 7538
**Subject**: Een web-based e-mail client in PHP5
**Template**: Normal.dot
**Title**: POP3ye
**Creation date**: 3/9/2004 8:35:00 PM
**Date**: 3/11/2004
**last Edit time**: 247
**last Print date**: 3/11/2004 2:01:00 AM
**Last save date**: 3/10/2004 5:15:00 PM
**Current time**: 2:01:51 AM

# 1 E-mail protocols

## 1.1 POP3

Het Post Office Protocol - versie 3 (POP3) werd ontwikkeld voor het tijdelijk opslaan van e-mail berichten op een server. Een POP3 server luistert standaard op TCP poort 110. Clients kunnen op deze poort een verbinding maken met de server en zo hun bewerkingen uitvoeren.

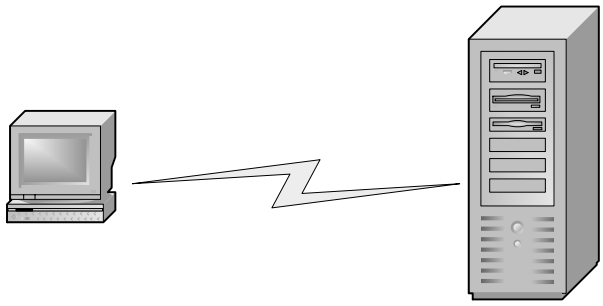
POP3 is niet bedoelt voor uitgebreide bewerkingen op de opgeslagen berichten. Het is de bedoeling dat e-mail van de server gedownload wordt en dan verwijdert.

Door deze reden beschikt POP3 dan ook over een beperkte instructieset.

Instructies die aan de POP3 server gegeven worden zijn case-insensitive en worden na hun argumenten beëindigt door CRLF (Carriage Return LineFeed of Enter). Alle POP3 instructies zijn 3 of 4 karakters lang. Argumenten worden van de instructie gescheiden door een spatie en kunnen 40 standaard ASCII karakters lang zijn.

Elke instructie die aan een POP3 server gegeven wordt wordt beantwoord met een statusindicator eventueel gevolgd door een boodschap van maximaal 512 karakaters (CRLF inclusief). Voor het ogenblik zijn er twee; negatief: '–ERR' en positief: '+OK' (altijd uppercase).

Het antwoord op sommige instructies kan meerdere regels omvatten; hierbij wordt elke regel beeindigd door een CRLF. De laatste regel van zo'n antwoord bevat een 'termination octet' (CRLF.CRLF).



Figuur 1: Communicatie tussen client en server via het TCP protocol.

Enkele user agents die POP3 ondersteunen:

- Microsoft Outlook & Outlook Express (win, mac)
- Mozilla Thunderbird (linux win)
- Kmail (linux)
- Pine (textbased)
- Mail (Mac)

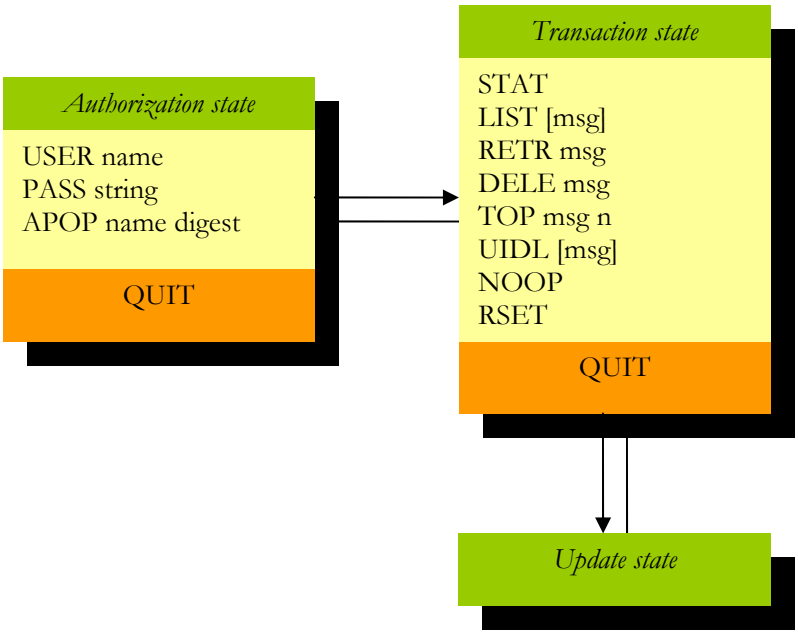
Enkele POP3 servers:

- Microsoft Exchange
- Mercur
- sendmail

## 1.1.1 De POP3 sessie

Een POP3 sessie doorloopt een aantal stappen nl.:

1. Authorization state
2. Transaction state
3. Update state




Figuur 2: Verschillende statussen van een POP3 sessie en hun mogelijke instructies.

### 1.1.1.1 Authorization state

Nadat een client op de server geconnecteerd is stuurt de server een welkomstgroet, hierna bevinden we ons in de *Authorization state*, de client moet zich nu identificeren dmv een username en password die beiden worden geauthenticeerd door de server. Als de juiste logindata verschaft wordt door de client wordt hij geauthorizeerd en kan hij overgaan naar de volgende stap; de *Transaction state*.

*Voorbeeld: (S = Server, C = Client)*

```
S: +OK pop s`maHegdelS
C: user jellepie
S: +OK 1320
C: pass mijnpasswd
S: +OK .. welcome
```

Zoals in het voorbeeld te zien is, wordt er na een successvolle verbinding met de POP3 server een welkomstboodschap getoond. Hierna start het authenticatieproces. De client geeft zijn username via de **USER** instructie, dit wordt gevolgd door een statusindicator van de server, idem voor de **PASS** instructie.

*Opmerking:De statusindicator die de server geeft bij het authorizatieproces is niet altijd juist. Doordat het mogelijk zou zijn alle mogelijke usernames af te gaan tot je een bestaande vindt, word bij sommige POP3 servers altijd een positieve statusindicator weergestuurd, ook bij onjuiste usernames.*

Aangezien het paswoord in plaintext over het netwerk gestuurd wordt kan dit een beveiligingsrisco met zich meebrengen. Daarom werd de **APOP** instructie ingevoerd.

#### 1.1.1.1.1 Authenticated POP

**APOP** is een alternatieve manier om op een POP3 server in te loggen. In plaats van via het klassieke **USER** / **PASS** authenticatiesysteem te werken geeft de server bij een connectie door een client een unieke welkomsboodschap. Deze boodschap wordt op de volgende manier opgebouwd:

```
+OK MSG <process-ID.clock@hostname>
```

De server stuurt een positieve statusindicator bij een connectie, eventueel gevolgd door een boodschap en hierna een unieke string opgebouwd uit:

de decimale notatie van het process-ID (PID)

de decimale waarde van de systeemklok

de FQDN van de host (fully qualified domain name)

De client onthoudt deze unieke string en hangt zijn paswoord er achter aan:

```
<process-ID.clock@hostname>paswoord
```

Hierna wordt op deze string het MD5 algoritme uitgevoerd, waardoor een checksum verkregen wordt die gebruikt wordt voor het inloggen.

```
S: +OK pop s`maHegdelS <1896.697170952@mail-in.pandora.be>
C: APOP jellepie b6e6b41cb770fd8f224396d46cc0da94
S: +OK .. welcome
```

In het voorbeeld ziet u de **APOP** instructie gevolgd door de username en de MD5 checksum als argumenten, waarna de user naar de *Transaction state* overgaat.

*Opmerking: Door gebruik van de **APOP** instructie wordt enkel het paswoord versleuteld over het netwerk verstuurd. Alle overige date wordt nog steeds in plaintext verzonden. Voor beveiliging van de overige data in e-mail kan gebruik gemaakt worden van PGPencryptie of een SSL verbinding.*

In de *authorization state* is het mogelijk de verbinding expliciet te verbreken door de **QUIT** instructie.

```
C: quit
S: +OK
```

### 1.1.1.2 Transaction state

Eens de client successvol geauthorizeerd is en de mailbox gelocked en geopent is op de server bevindt de POP3 sessie zich in de *transaction state*. De client kan de server nu verschillende instructies geven mbt het ophalen en wissen van e-mail.

Nadat de client zijn laatste instructie gegeven heeft kan de POP3 sessie beëindigt worden door de **QUIT** instructie.

De instructies die gebruikt kunnen worden zijn:

- **STAT**
  - Vraagt aan de server hoeveel berichten er in de mailbox zitten gevolgd door de totale grootte van alle berichten. Ook wel 'drop listing' genoemd (mailbox = maildrop).

    - ```
      C: stat
      S: +OK 3 9030
      ```

      Betekent dat er 3 berichten in de mailbox zit met een grootte van 9030 bytes.

      Geavanceerde implementaties van POP3 kunnen meer informatie weergeven bij de **STAT** instructie.

- **LIST [msg]**
  - Als geen argument gegeven wordt volgt er een lijst van alle e-mail in de mailbox, waarbij elke regel een 'scan listing' voor de desbetreffende email wordt genoemd. Als er wel een argument meegegeven is volgt enkel de scan listing voor de gevraagde e-mail.

    - ```
      C: list 1
      S: +OK
         1 2569
      ```

      **LIST** instructie met als parameter het berichtnummer toont de scan listing van de betreffende e-mail.

    - ```
      C: list
      S: +OK
          1 2569
          2 5874
          3 587
          .
      ```

      **LIST** instructie zonder parameters elk bericht wordt op een afzonderlijke regel weergegeven met een uniek nummer en de grootte, gescheiden door een enkele spatie tussen nummer en grootte en een CRLF tussen elke regel. De lijst wordt beëindigd door een punt.
      ```

- **RETR msg**
  - Stuurt de inhoud van de e-mail met ID msg weer en eindigt met de termination octet.

```
C: retr 1
S: +OK 26432 octets
Return-Path: <carrierepush@vacaturemedia.com>
Delivered-To: jellepie@pandora.be
Received: (qmail 8686 invoked from network); 12 Feb 2004 18:21:32 -0000
Received: from unknown (HELO sirios.telenet-ops.be) ([195.130.132.52])
          (envelope-sender <carrierepush@vacaturemedia.com>)
          by menthe.telenet-ops.be (qmail-ldap-1.03) with SMTP
          for <jellepie@pandora.be>; 12 Feb 2004 18:21:32 -0000
Received: from 127.0.0.1 (localhost [127.0.0.1])
        by sirios.telenet-ops.be (Postfix) with SMTP id 9543F3BE21
        for <jellepie@pandora.be>; Thu, 12 Feb 2004 19:21:32 +0100 (MET)
Received: from VACATURE.COM (host036.be.dedigate.com [212.3.247.36])
        by libya.telenet-ops.be (Postfix) with SMTP id 80E5A37E71
        for <jellepie@pandora.be>; Thu, 12 Feb 2004 19:21:32 +0100 (MET)
Message-Id: <402bc332.25@VACATURE.COM>
From: Vacature.com - CarriÞrePush <carrierepush@vacaturemedia.com>
To: Piekaerts Jelle <jellepie@pandora.be>
Subject: CarriÞrePush: de valkuilen van solliciteren via e-mail
Date: Thu, 12 Feb 2004 19:17:22 +0200
Reply-To: <carrierepush@vacaturemedia.com>
X-MA-Reference: SiUmHn83SSL47VSzdxt8SSO
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="----=_NexPart_000"

...

<TD><IMG height=3D5 src=3D=22http://www.vacature.com/images/carrierep=
ush/blanc.gif=22 width=3D436></TD></TR></TBODY></TABLE></TD></TR></TBO=
DY></TABLE><IMG width=3D'1px' height=3D'1px' SRC=3D'http://212.3.227.3=
2/optiext/optiextension.dll?ID=3Dr1tXCWbkrrUUUUUVeuHHrrd'></BODY></HTM=
L>
------=_NexPart_000--
```

.

- **DELE msg**
  - Markeert een bericht msg om gewist te worden, het eigenlijke wissen van het bericht gebeurt pas in de *update state*. Elke referentie naar deze berichten wordt eveneens ongeldig.

    - ```
      list
      +OK
      1 2358
      .
      dele 1
      +OK
      list
      +OK
      .
      list 1
      -ERR already deleted
      dele 1
      -ERR already deleted
      retr 1
      -ERR already deleted
      ```

- **NOOP**
      ```

- De **NOOP** instructie doet niets, de server geeft een positieve statusindicator. Kan gebruikt worden om de connectie tss de client en server te behouden als er niets gedaan wordt.

  ```
  C: noop
  S: +OK
  ```

- **RSET**
  - Zoals reeds vermeld zal de **DELE** instructie de te wissen berichten enkel markeren. De **RSET** instructie kan deze markering ongedaan maken.

- **QUIT**
  - Brengt de POP3 sessie naar de *update state*.

- **TOP msg n**
  - De eerste n regels van de body van het bericht met id msg worden weergestuurd. Indien de body korter is als n wordt het hele bericht gestuurd. De header wordt automatisch getoond. Werkt zoals **RETR** maar met een parameter meer.
- **UIDL [msg]**
  - Unique ID Listing. Geeft een lijst van elk bericht met een UID. Dit is nuttig wanneer berichten niet direct van de server verwijdert worden door de client. De client kan het UID bijhouden en op basis hiervan enkel de nieuwe berichten downloaden.

    ```
    uidl
    +OK
    1 1076613662.19699.menthe.telenet-ops.be,S=2356
    2 1076613662.19761.menthe.telenet-ops.be,S=2388
    3 1076613666.20392.menthe.telenet-ops.be,S=2350
    .
    uidl 1
    +OK 1 1076613662.19699.menthe.telenet-ops.be,S=2356
    ```

    Het gebruik van **UIDL** en het negeren van de **DELE** instructie kan gebruikt worden als semi-permanente opslag voor gegevens (berichten).

*Opmerking: Berichten die gemarkeerd zijn om te wissen, zullen niet meer weergegeven worden in het resultaat van instructies noch mag hun id gebruikt worden als argument van een instructie. Indien wel zal er een –ERR bericht weergegeven worden.*

### 1.1.1.3 Update state

Als de client de **QUIT** instructie geeft in de transaction state komt de POP3 sessie in de *update state*. Als de **QUIT** instructie gegeven wordt vanaf de *authorisation state* zal de *update state* niet bereikt worden.

Berichten die gemarkeerd werden voor verwijdering door de **DELE** instructie worden nu verwijdert en de verbinding tussen de client en server wordt beëindigt.

## 1.2 SMTP

SMTP staat voor Simple Mail Transfer Protocol, een protocol voor het efficiënt en betrouwbaar verzenden van e-mail. Efficiënt als in: een bericht met meerdere ontvangers hoeft niet meerdere malen verzonden te worden, waardoor bandbreedte bespaard wordt. En betrouwbaar als in: alle input die aan een SMTP server gegeven word, wordt geverifieerd vooraleer goedgekeurd.

SMTP is niet specifiek gebonden aan het TCP protocol. Een belangrijke eigenschap is het verzenden van e-mail over verschillende netwerken, dit noemt ment "SMTP mail relaying", deze netwerken worden onderling verbonden door een relay of gateway (smtp gateway tss netwerken die verschillende protocols gebruiken).

SMTP kan gebruikt worden tussen SMTP servers onderling - waarbij één server SMTP client is en de andere de SMTP server - evanals tussen een gewone SMTP client en een SMTP Server.

Tussen verzender en ontvanger kan een e-mail bericht dus langs verschillende relays/gateways reizen. De MX (Mail eXchanger) records van het DNS (Domain Name System) worden gebruikt om de volgende hop van het bericht te bepalen.

Een e-mail bericht bevat een 'envelope' en 'content'.

De envelope bestaat uit het e-mail adres van de afzender, naar wie ook eventuele foutmeldingen worden gestuurd (bvb. als een mailbox vol zit) alsook de bestemmelingen.

Het content gedeelte wordt gestuurd in het SMTP **DATA** gedeelte en bestaat uit twee delen; de headers en het body. De headers zijn een collectie van velden met hun respectievelijke waarde, in US-ASCII. Het body is tekstueel in US-ASCII maar kan ook in MIME zijn.

Elke SMTP instructie of **DATA** lijn wordt beindigd door een CRLF. De lengte van een lijn kan beperkt worden door de server.

SMTP instructies zijn case insensitive, tenzij anders aangegeven door service extensions (zie ESMTP).

De standaard poort waarop een SMTP server draait is 25

### 1.2.1 Enhanced SMTP

De mogelijkheid bestaat het SMTP protocol buiten de vastgelegde standaarden uit te breiden dmv zogenaamde 'SMTP service extensions'. De client en server moeten dan wel beide deze mogelijkheden ondersteunen. SMTP met service extensions wordt Enhanced SMTP of ESMTP genoemd.

Om backward compatibility te waarborgen moeten de vastgelegde standaarden ook steeds nageleefd worden.

Met de komst van de SMTP service extensions werd ook de nieuwe **EHLO** instructie ingevoerd, die als een server smtp extensions ondersteund, deze ook weergeeft. Ook worden alle extensions bijgehouden in een register waarbij hun wijzigingen aan het **MAIL RCPT** en **DATA** instructies worden bewaard, evenals andere gegevens betreffende de extension. Dit register wordt beheerd door het IANA (Internet Assigned Numbers Authority: http://www.iana.org/).

Een voorbeeld van een service extension is AUTH (RFC2554), deze service maakt het mogelijk een client zich te laten authoriseren vooraleer hij de ESMTP server mag gebruiken. Standaard is authorisatie niet noodzakelijk. (http://www.iana.org/assignments/mail-parameters).

Een ander voorbeeld is de **SIZE** extension. **SIZE** kan als argument gebruikt worden bij de **MAIL FROM** instructie. De client geeft de size aldus mee en als deze de server zijn vastgelegde maximum verzendbare bericht overschrijdt zal dit gemeld worden. Aldus zal het bericht niet verzonden worden en word er kostbare bandbreedte bespaard.

Een van de belangrijkste extensions is **8BITMIME**. Standaard US-ASCII is 7-bit maar tegenwoordig, worden alle attachments als 8-bit **MIME** geencodeerd. Deze extension is dus een noodzaak.

## 1.2.2 De ESMTP Sessie

Een SMTP sessie start wanneer een client verbindt met een server. Een client kan gewoon een pure telnet sessie zijn als een UA zoals Outlook, Kmail, etc.

Een smtp sessie moet niet altijd geinitialiseerd worden, enkele instructies kunnen gebruikt worden zonder initialisatie. Deze zijn **VRFY**, **EXPN**, **NOOP** en **QUIT**.

Na elke instructie door de client antwoord de server met een statuscode die bestaat uit drie cijfers. Deze codes worden gebruikt voor error handling, debugging en als gewone bevestiging, ze kunnen optioneel ook gevolgd worden door een boodschap. Door het gebruik van deze codes wordt SMTP een betrouwbaar protocol. Naar gelang de reply code kan de client al dan niet actie ondernemen.

Elke lijn die een antwoord op eenzelfde instructie bevat moet voorafgegaan worden door dezelfde driecijferige reply code gevolgd door het '-' teken. Behalve de laatste lijn.

### 1.2.2.1 Initialisatie

Eens de client succesvol verbonden is stuurt de server een welkomstbericht naar de client. Hierna stuurt de client de instructie **HELO** of **EHLO** om zich te identificeren, gevolgd door een FQDN als de client hierover beschikt. Zoals eerder vermeldt is de **EHLO** instructie geimplemnteerd voor service extenions, maar bleef **HELO** bewaard voor backward compatiblity.

ESMTP:

```
S: 220 apate.telenet-ops.be ESMTP Postfix
C: ehlo Jelle
S: 250-apate.telenet-ops.be
S: 250-PIPELINING
S: 250-SIZE 5120000
S: 250-VRFY
S: 250-ETRN
S: 250-XVERP
S: 250 8BITMIME
```

SMTP:

```
220 apate.telenet-ops.be ESMTP Postfix
helo Jelle
250 apate.telenet-ops.be
```

### 1.2.2.2 Mail transaction

De eerste instructie die een client na **EHLO** geeft is **MAIL FROM**, waarmee aangegeven wordt wie de afzender van de e-mail is, ook reverse-path genoemd.

```
MAIL FROM:<reverse-path> <CRLF>
```

Vb:

```
MAIL FROM: <jelle.piekaerts@pandora.be>
S: 250 Ok
```

Het reverse path moet fully qualified zijn, het moet dus voldoen aan de eisen van de compositie van een e-mail adres. Als er niets als reverse path opgegeven wordt spreekt men over een null-reverse path. (niets -> lees: <>).

Eens de **MAIL FROM** instructie gegeven is worden de status en de buffers van de SMTP sessie gereset.

De volgende stap is het opgeven van het adres van de ontvanger, het zogenaamde forward-path. Dit gebeurt via de **RCPT TO** instrcutie. Deze instructie dient herhaald te worden naar gelang het aantal forward-paths.

```
RCPT TO:<forward-path> <CRLF>
```

Vb:

```
C: RCPT TO: jelle.piekaerts@pandora.be
S: 250 Ok
```

De volgorde waarin de instructies gegeven worden is van belang. Indien de instructies niet in de correcte volgorde volgen zal een '503 Bad sequence of commands' error volgen.

Nadat de envelope van het bericht opgesteld is, volgt het ingeven van de eigenlijke content van het bericht via de **DATA** instrcutie. De **DATA** instrcutie dient afgesloten te worden door <CRLF>.<CRLF>.

Mogelijke memo headers die gebruikt kunnen worden zijn; date, subject, to, from, cc, …. Deze headers zijn steeds in US-ASCII vorm.

```
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: subject: email title
C:
C: Hello, this is an email
c: .
S: 250 Ok: queued as A9FF637E93
```

Nadat het **DATA** gedeelte gedaan is kan de client nog een e-mail sturen door opnieuw het commando **MAIL TO**, **EHLO** / **HELO** of **RSET** te gebruiken. De status en buffers (reverse-path, forward-path en data buffer) worden hierbij gewist. Anderzijds kan de client ook afsluiten door de **QUIT** instructie.

De **RSET** instructie kan op elk ogenblik tijdens de sessie gegeven worden, en verwijdert zoals eerder vermeld alle statussen en buffers, **HELO** kan hiervoor ook gebruikt worden maar deze instructie brengt extra overhead met zich mee.

**RSET**, **QUIT** en **DATA** aanvaarden geen parameters

**NOOP** doet niets buiten de verbinding levend te houden, **QUIT** daarentegen zorgt ervoor dat de verbinding tussen client en server gesloten wordt.

Enkele restricties ivm lengte en grootte:

- Local-part of username → max 64chars
- Domain name of number → max 255 chars
- Reverse of forward path → max 256 chars (inclus scheidingen @ . …)
- Instructie en params → max 512 chars
- Reply lijn → max 512 chars
- Tekst lijn → 1000 chars (can be extended by extensions service)
- Compleet bericht → het min toegeleten max 64k (SIZE service extension)
- Recipients buffer → min toegelate max = 100 recipients

### 1.2.2.3 Een eenvoudige ESMTP sessie

```
S: 220 adicia.telenet-ops.be ESMTP Postfix
C: ehlo jelle
S: 250-adicia.telenet-ops.be
S: 250-PIPELINING
S: 250-SIZE 5120000
S: 250-VRFY
S: 250-ETRN
S: 250-XVERP
S: 250 8BITMIME
C: mail from: <jellepie@pandora.be>
S: 250 Ok
C: rcpt to: ikke@pandoa.be
S: 450 <ikke@pandoa.be>: Recipient address rejected: Domain not found
C: rcpt to: ikke@pandora.be
S: 250 Ok
C: rcpt to: jellepie@pandora.be
S: 250 Ok
C: rcpt to: jelle.piekaerts@pandora.be
S: 250 Ok
C: rcpt to: k-rash@pandora.be
S: 502 Error: command not implemented
C: rcpt to: krash@pandora.be
S: 250 Ok
C: rcpt to: k-rash@pandora.be
S: 250 Ok
C: rct
S: 502 Error: command not implemented
C: rcpt to : joske@microsoft.com
```

```
S: 501 Syntax: RCPT TO: <address>
C: data
S: 354 End data with <CR><LF>.<CR><LF>
C: subject: dit is het onderwerp
C: to: alleman
C: from: jelleman
C:
C: dit is een bericht voor alleman van jelleman
C:
C:.
S: 250 Ok: queued as 474521F7113
C: quit
S: 221 Bye
```

### 1.2.3 Analyse van een e-mail header

```
Return-Path: <henri.jans@pandora.be>
Delivered-To: jelle.piekaerts@pandora.be
Received: (qmail 26731 invoked from network); 17 Feb 2004 17:38:47 -0000
Received: from unknown (HELO dionysos.telenet-ops.be) ([195.130.132.51])
          (envelope-sender <henri.jans@pandora.be>)
          by menthe.telenet-ops.be (qmail-ldap-1.03) with SMTP
          for <jelle.piekaerts@pandora.be>; 17 Feb 2004 17:38:47 -0000
Received: from 127.0.0.1 (localhost [127.0.0.1])
       by dionysos.telenet-ops.be (Postfix) with SMTP
       id E51BB405FC6; Tue, 17 Feb 2004 18:38:46 +0100 (MET)
Received: from elpis.telenet-ops.be (elpis.telenet-ops.be [195.130.132.40])
       by ladon.telenet-ops.be (Postfix) with ESMTP
       id ACC5722C015; Tue, 17 Feb 2004 18:38:46 +0100 (MET)
Received: from localhost (adicia.telenet-ops.be [195.130.132.56])
       by elpis.telenet-ops.be (Postfix) with SMTP
       id 9ED0B37F24; Tue, 17 Feb 2004 18:38:46 +0100 (MET)
Received: from bart (D5772138.kabel.telenet.be [213.119.33.56])
       by adicia.telenet-ops.be (Postfix) with SMTP
       id 5A7131F7111; Tue, 17 Feb 2004 18:38:46 +0100 (MET)
Message-ID: <001101c3f57c$e74d2b90$382177d5@bart>
From: "Bart Jans" <henri.jans@pandora.be>
To: "De Strijcker Laurens" <laurens.de.strijcker@pandora.be>,
       "Callebaut Kevin" <callebauter@skynet.be>, "Ralf" <ralf@pandora.be>,
       "Jelle Piekaerts" <jelle.piekaerts@pandora.be>
Subject: Fw: njam njam
Date: Tue, 17 Feb 2004 18:38:50 +0100
MIME-Version: 1.0
Content-Type: multipart/mixed;
       boundary="----=_NextPart_000_000E_01C3F585.48F754D0"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
```

Wanneer een SMTP server een e-mail bericht ontvangt voor aflevering of te relayen, moet hij steeds zijn gegevens aan het begin van het bericht toevoegen (FQDN en IP). Dit wordt een e-mail trace genoemd. De server die het bericht uiteindelijk aflvert voegt er het return-path aan toe voor eventuele problemen met het bericht te melden.

Na de trace volgt de envelope en eventueel andere headers, specifiek aan de UA of de encoding van het bericht.

### 1.2.4 Security

Gedurende de voorbij jaren is het gebruik van de relay mogelijkheid, die SMTP heeft misbruikt om de originele oorsprong van e-mail te verbergen. Daardoor wordt relaying nu door vele servers enkel toegestaan door hosts die door de SMTP server bekend zijn.

Ook zijn de replies die op de **VRFY** en **EXPN** instructie gegeven worden niet altijd meer correct, om de privacy van gebruikers te beschermen en misbruik te voorkomen.

Het SMTP protocl bevat standaard geen beveiliging en of authorisatie. Evanals wordt alles in plaintext verzonden over het netwerk. Om auhtorisatie in te voeren moeten er service extensions gebruikt worden. Voor het bericht te beveiligen kan het geencodeerd worden met het PGP protocl, of kan er een SSL verbdining gebruikt worden.

## 1.3 IMAP

Het IMAP4rev1 protocol, stelt een client in staat om e-mail berichten te bekijken en manipuleren op een IMAP4rev1 server.

IMAP4rev1 laat toe 'mappen met berichten' of 'mailboxes' die op een remote server opgeslagen zijn te bewerken op een wijze die vergelijkbaar is met lokale e-mail client programma's.

Het is mogelijk om te synchroniseren tussen een lokaal e-mail programma met ondersteuning voor IMAP4rev1 en de IMAP4rev1 server (IMAP-DISC).

Mailboxen kunnen gemaakt, verwijdert en hernoemt worden. Berichten in deze mailboxen worden ingedeeld in folders.

Er kan gecontroleerd worden op nieuwe berichten, deze kunnen permanent verwijdert worden. Flags kunnen gezet en verwijdert worden (rfc-822) en MIME-IMV berichten kunnen geparsed worden. Individuele eigenschappen van berichten kunnen opgevraagd worden (headers, attributen, tekst, …).

Elk bericht krijgt een uniek nummer en kan hiermee aangesproken worden. (sequentiele nummering of een UID: unique identifier)

IMAP4rev1 voorziet niet in een manier voor het verzenden van e-mail, hiervoor dient het SMTP protocol.

Standaard luistert een IMAP4rev1 server op poort 143.

Elke instructie die een client aan de server geeft moet vooraf gegaan worden door een unieke alfanumerieke identifier, 'tag' genoemd (vb. ABC001, A00001, … ) en wordt beeindigd door CRLF. De server gebruikt dit tag als prefix voor zijn antwoord.

De bedoeling van een tag is tweezijdig; ten eerste zo kan de client meerdere instructies sturen zonder te wachten op antwoord van de server, hij kan dan afleiden adhv het prefix op welke instructie de server antwoord.

Als de server data doorstuurt of informatie die niet als antwoord op een instructie bedoelt is wordt dit voorafgegaan door een '*'.

Een server kan drie mogelijke antwoorden geven op een instructie; OK, NO of BAD:

- OK: de instructie is succesvol uitgevoerd;
- NO: de instructie is niet succesvol uitgevoerd;
- BAD: protocol error of syntaxfout.

De client onderneemt actie naargelang het antwoord.

Elk bericht in een IMAP4rev1 mailbox heeft enkele attributen, deze kunnen onafhankelijk van elkaar opgevraagd worden.

- Sequentieel bericht nummer: van 1 tot het aantal berichten in de mailbox;
- UID attribuut: het unieke nummer van een bericht (dit nr is niet noodzakelijk sequentieel maar wel incrementeel);
- Message flags: kunnen permanent of voor de duur van een session zijn
    - System flags beginnen met een '\' (\Seen, \Answered, \Flagged, \Deleted, \Draft, \Recent);
    - Flags kunnen ook door een client gedefinieerd worden.
- Date & time (wanneer het bericht toegekomen is op server);
- Bericht grootte;
- Envelope structuur;
- Bericht body, header, MIME header of onderdeel;

### 1.3.1 De IMAP4rev1 sessie

Een IMAP4rev1 sessie kan zich in 4 statussen bevinden. De meeste IMAP4rev1 instructies zijn echter alleen geldig in één bepaalde status. Als een instructie gegeven wordt in een status die het niet ondersteunt volgt een NO of BAD antwoord van de server.

Enkele instructies kunnen gegeven worden in alle statussen, deze zijn **CAPABILITY**, **NOOP** en **LOGOUT**.

De **CAPABILITY** instructie vraagt de server een lijst van wat hij ondersteunt, hierin moet zeker IMAP4rev1 voorkomen als minumum.

```
C: abcd CAPABILITY
S: * CAPABILITY IMAP4rev1 AUTH=KERBEROS_V4
S: abcd OK CAPABILITY completed
```

*Zoals te zien in het bovenstaande voorbeeld antwoord de server met een untagged antwoord waaruit blijkt dat hij Kerberos v4 authenticatie ondersteunt.*

De **NOOP** instructie doet niets buiten de verbinding levend te houden (autologout timer resetten) en kan ook gebruikt worden voor de status te updaten; een IMAP4rev1 server kan op elk ogenblik informatie sturen die een client niet gevraagd heeft (bvb. als er een nieuw bericht op de server toegekomen is, als er flags verandert zijn, …).

```
C: a002 NOOP
S: a002 OK NOOP completed
. . .
C: a047 NOOP
```

```
S: * 22 EXPUNGE
S: * 23 EXISTS
S: * 3 RECENT
S: * 14 FETCH (FLAGS (\Seen \Deleted))
S: a047 OK NOOP completed
```

*Zoals in het bovenstaande voorbeeld te zien werd er na de **NOOP** instructie untagged status informatie meegestuurd door de server. Deze informatie zal uiteraard alleen gestuurd worden als men zich in de selected state bevindt.*

**LOGOUT** zal de verbinding met de server verbreken. De server antwoord met een untagged BYE gevolgd door een tagged OK.

```
C: A023 LOGOUT
S: * BYE IMAP4rev1 Server logging out
S: A023 OK LOGOUT completed
```

### 1.3.1.1 Non-authenticated state

In deze status moet de client zich auhtoriseren. Een beperkt aantal instructies is mogelijk in deze staat. Deze status wordt bereikt wanneer een client met een server verbindt. Als een client al pre-geauthoriseerd is zal hij onmiddellijk naar de authenticated state gaan.

In deze status kunnen de **AUTHENTICATE** en **LOGIN** instructie gebruikt worden om over te gaan naar de authenticated state.

**AUTHENTICATE** kan gebruik maken van een aantal authenticatie technieken (een server geeft enkele challenges waarop de client antwoord). **LOGIN** gebruikt de traditionele plaintext user / paswoord manier.

```
S: * OK KerberosV4 IMAP4rev1 Server
C: A001 AUTHENTICATE KERBEROS_V4
S: + AmFYig==
C: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4DT
+nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
WwuQ1MWiy6IesKvjL5rL9WjXUb9MwT9bpObYLGOKi1Qh
S: + or//EoAADZI=
C: DiAF5A4gA+oOIALuBkAAmw==
S: A001 OK Kerberos V4 authentication successful
C: a001 LOGIN SMITH SESAME
S: a001 OK LOGIN completed
```

*Voorbeeld van de twee authenticatiemogelijkheden.*

### 1.3.1.2 Authtenticated state

De client is geauthenticeerd en moet nu een mailbox selecteren waarop hij bewerkingen zal uitvoeren.

Deze status wordt onmiddellijk bereikt als de client pre-geauthoriseerd is.

De **SELECT** en **EXAMINE** instructies worden gebruikt om de mailbox te openen en de selected state te bereiken.

Volgende instructies zijn ook geldig in deze status: **CREATE**, **DELETE**, **RENAME**, **SUBSCRIBE**, **UNSUBSCRIBE**, **LIST**, **LSUB**, **STATUS**, en **APPEND**.

De **SELECT** instructie wordt gevolgd door de naam van de mailbox. De server zijn antwoord hierop omvat:

- * FLAGS (flags): de bestaande flags;
- * <n> EXISTS: het aantal berichten;
- * <n> RECENT: het aantal nieuwe berichten (\Recent flag set);
- * OK [UIDVALIDITY <n>]: de UID geldigheidswaarde.
- * OK [<n> UNSEEN] : niet bekeken berichten (optioneel)
- * OK [PERMANENTFLAGS (flags )]: als de client enkele flags niet kan wijzigen (optioneel)
- [READ-WRITE] of [READ-ONLY] in server tagged antwoord.

```
C: A142 SELECT INBOX

S: * 172 EXISTS
S: * 1 RECENT
S: * OK [UNSEEN 12] Message 12 is first unseen
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
S: A142 OK [READ-WRITE] SELECT completed
```

*Uit de bovenstaande SELECT instructie kan men afleiden dat in de inbox mailbox:*

- *172 berichten zitten;*
- *waarvan 1 nieuw*
- *waarvan 12 niet bekeken beginnend bij bericht 12*
- *3857529045 geldige UIDs*
- *De volgende flags beschikbaar zijn:*
  - *\Answered*
  - *\Flagged*
  - *\Deleted*
  - *\Seen*
  - *\Draft*
- *De flags \Deleted \Seen en alle overige zijn permanent dus niet op sessie basis*

De **EXAMINE** instructie is hetzelfde als **SELECT** maar de mailbox wordt geopent in READ-ONLY modus. De server zal dus antwoorden met een READ-ONLY, er kunnen geen permanente wijzigen gemaakt worden daardoor dus ook geen PERMANENTFLAGS.

Voorbeeld:

```
C: A932 EXAMINE box
S: * 17 EXISTS
S: * 2 RECENT
S: * OK [UNSEEN 8] Message 8 is first unseen
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS ()] No permanent flags permitted
S: A932 OK [READ-ONLY] EXAMINE completed
```

**CREATE** zal een nieuwe mailbox aanmaken, tenzij deze al bestaat.

Voorbeeld:

```
C: A003 CREATE News/
S: A003 OK CREATE completed
C: A004 CREATE News/Music
S: A004 OK CREATE completed
```

Meerdere mailboxen kunnen tegelijk gemaakt worden door het gebruik van de hierarchische scheider. ('/'). Deze kan opgevraagd worden met de **LIST** instructie.

**DELETE** is de tegenhanger van **CREATE** en zal een bestaande mailbox permanent wissen. Een mailbox met inferieure mailboxen EN het \Noselect attribuut mag niet gewist worden. Bvb. in mailbox/news/music mag mailbox of news niet gewist worden, als voor mailbox het \Noselect attribuut van toepassing is, indien het \Noselect attribuut nvt is mag dit wel gewist worden.

Voorbeeld:

```
C: A682 LIST "" *
S: * LIST () "/" blurdybloop
S: * LIST (\Noselect) "/" foo
S: * LIST () "/" foo/bar
S: A682 OK LIST completed
C: A683 DELETE blurdybloop
S: A683 OK DELETE completed
C: A684 DELETE foo
S: A684 NO Name "foo" has inferior hierarchical names
C: A685 DELETE foo/bar
S: A685 OK DELETE Completed
C: A686 LIST "" *
S: * LIST (\Noselect) "/" foo
S: A686 OK LIST completed
C: A687 DELETE foo
S: A687 OK DELETE Completed
```

Opmerking: Als '* LIST (\Noselect) "/" foo' - '* LIST () "/" foo' geweest was had de DELETE foo instructie wel gewerkt.

Het hernoemen van een mailbox kan door het gebruik van de **RENAME** instructie. Als een mailbox met één of meerdere inferieure mailboxen hernoemt wordt worden alle sub-mailboxen gerenamed.

Voorbeeld:

```
C: A682 LIST "" *
S: * LIST () "/" blurdybloop
S: * LIST (\Noselect) "/" foo
S: * LIST () "/" foo/bar
S: A682 OK LIST completed
C: A683 RENAME blurdybloop sarasoop
S: A683 OK RENAME completed
C: A684 RENAME foo zowie
S: A684 OK RENAME Completed
C: A685 LIST "" *
S: * LIST () "/" sarasoop
S: * LIST (\Noselect) "/" zowie
S: * LIST () "/" zowie/bar
S: A685 OK LIST completed
```

# LIST EN SUBSCRIBE EN UNSUBSCRIBE & LSUB nog meer over zoeken

De **SUBSCRIBE** en **UNSUBSCRIBE** instructies worden gebruikt om een mailbox toe te voegen / subscriben bij de actieve mailboxes op de server.

**LIST** geeft een deel van de namen weer van van het geheel waartoe een client toegang heeft. Nul of meer untagged antwoorden kunnen worden weergestuurd door de server, samen met hun naam, attributen, hierarchische scheider (delimiter).

Een lege string "" als reference argument betekent dat de mailbox genomen wordt die gespecificeerd werd door de **SELECT** instructie.

De **STATUS** instructie vraagt de status van de geselecteerde mailbox, deze instructie wijzigt niets aan de mailbox. De statusinformatie die kan opgevraagd worden is:

- MESSAGES: aantal berichten in de mailbox
- RECENT: aantal berichten met de \Recent flag
- UIDNEXT: UID dat zal gegeven worden aan het volgende nieuw bericht
- UIDVALIDITY: de uid value van de mailbox
- UNSEEN: berichten die de \Seen flag niet hebben

Voorbeeld:
```
C: A042 STATUS blurdybloop (UIDNEXT MESSAGES)
S: * STATUS blurdybloop (MESSAGES 231 UIDNEXT 44292)
S: A042 OK STATUS completed
```

**APPEND** voegt een nieuw bericht toe aan het einde van een bepaalde mailbox. Het formaat dient in RFC-822 vorm te zijn.

Example:
```
C: A003 APPEND saved-messages (\Seen) {310}
C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
C: From: Fred Foobar <foobar@Blurdybloop.COM>
C: Subject: afternoon meeting
C: To: mooch@owatagu.siam.edu
C: Message-Id: <B27397-0100000@Blurdybloop.COM>
C: MIME-Version: 1.0
C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
C:
C: Hello Joe, do you think we can meet at 3:30 tomorrow?
C:
S: A003 OK APPEND completed
```

## 1.3.1.3 Selected state

Eens een mailbox succesvol geslecteerd is bevinden we ons in de selected state. In deze staat kunnen de berichten die zich in een mailbox bevinden gemanipuleerd worden.

De instructies die in alle statussen geldig zijn en die van de authenticated state zijn hier ook nog steeds geldig. Daarboven komen nog enkele nieuwe instructies die enkel geldig zijn in de slected state:

**CHECK**, **CLOSE**, **EXPUNGE**, **SEARCH**, **FETCH**, **STORE**, **COPY**, and **UID**.

**CHECK** is gelijk aan de **NOOP** instructie en kan dus gebruikt worden om de status van de mailbox en berichten na te gaan.

De **CLOSE** instructie verwijdert alle berichten uit de geselecteerde mailbox die de \Deleted flag hebben en keert daarna weer naar de authenticated state vanuit de selected state.

**EXPUNGE** verwijdert alle berichten die de \Deleted flag hebben, waarbij de server voor elk verwijdert bericht een regel geeft.

Voorbeeld:
```
S: A202 EXPUNGE
S: * 3 EXPUNGE
S: * 3 EXPUNGE
S: * 5 EXPUNGE
S: * 8 EXPUNGE
S: A202 OK EXPUNGE completed
```

Het IMAP4rev1 protocol bezit een zeer krachtige zoekfunctie die gebruikt kan worden voor het zoeken van berichten in de mailbox die aan bepaalde voorwaarden voldoen. De instructie die hiervoor gebruikt wordt is **SEARCH**.

Voorbeeld:
```
C: A282 SEARCH FLAGGED SINCE 1-Feb-1994 NOT FROM "Smith"
S: * SEARCH 2 84 882
S: A282 OK SEARCH completed
```

De **FETCH** instructie haalt een of meerdere attributen van een bericht op, waarentegen **STORE** flags van een bericht kan wijzigen.

Voorbeeld:
```
C: A654 FETCH 2:4 (FLAGS BODY[HEADER.FIELDS (DATE FROM)])
S: * 2 FETCH ....
S: * 3 FETCH ....
S: * 4 FETCH ....
S: A654 OK FETCH completed
```

Voorbeeld:
```
C: A003 STORE 2:4 +FLAGS (\Deleted)
S: * 2 FETCH FLAGS (\Deleted \Seen)
S: * 3 FETCH FLAGS (\Deleted)
S: * 4 FETCH FLAGS (\Deleted \Flagged \Seen)
S: A003 OK STORE completed
```

Via de **COPY** instructie kan één of meer berichten van een bepaalde mailbox naar een andere kopieren.

Voorbeeld:
```
C: A003 COPY 2:4 MEETING
S: A003 OK COPY completed
```

De **UID** instructie kan 2 vormen aannemen. In de eerste vorm, kan het argumenten als **COPY**, **FETCH** en **STORE** aannemen met hun respectievelijke argumenten. De nummers voor de berichten zijn nu UID nummers ipv sequentiële nummers.

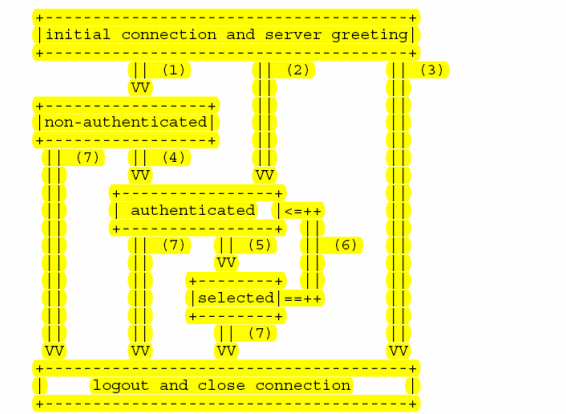
In de tweede vorm neemt **UID** een **SEARCH** instructie als argument met zijn respectievelijke argumenten, met het verschil dat er nu geen sequentiele nummers als antwoord komen maar UIDs.

Voorbeeld:
```
C: A999 UID FETCH 4827313:4828442 FLAGS
S: * 23 FETCH (FLAGS (\Seen) UID 4827313)
S: * 24 FETCH (FLAGS (\Seen) UID 4827943)
S: * 25 FETCH (FLAGS (\Seen) UID 4828442)
S: A999 UID FETCH completed
```

*Opmerking: Het is mogelijk een bereik van UIDs te gebruiken bij instructies die dit ondersteunen, maar er is geen garantie dat de berichten elkaar echt opvolgen, zoals bij sequentiele nummering.*

### 1.3.1.4 Logout state

In de logout state wordt de verbinding tussen C en S verbroken. Dit kan door de C gestart zijn of door de S.




## 1.4 Beveiliging

Standaard voorzien dezez protocols weinig tot geen bescherming van de informatie die verzonden en/of ontvangen wordt. De beveiligingsmogelijkheden die wel vooezien zijn worden in de praktijk ook niet altijd benuttigd.

### 1.4.1 SSL / TLS

SMTP, POP3

### 1.4.2 PGP

Enige mogelijkheid die volledige veiligheid kan bieden.

# 2 MIME

Multipurpose internet mail extensions

# 3 Functionaliteiten

- **Registration before usage**: on / off
  - Off: Users kunnen zonder registratie een POP3 account checken.
  - On: Users moeten zich registreren alvorens ze gebruik kunnen maken van POP3ye.

    - Gebruik maken van zo'n On / Off setting zou databank gebruik helemaal kunnen uitsluiten, wat betekent dat er geen settings of data kan opgeslagen worden.
    - De user geeft zijn login / passwd en servernaam eventueel met SSL, POP3ye haalt berichten op en geeft ze weer.

- **Multiple accounts**: on / off / n (global)
  - Mag een user meer dan één accnt maken.
    - ~~On: ja, onbeperkt.~~
    - ~~Off: nee, slechts één~~.
    - N: ~~ja~~, n.

- **Listing:**
  - Adressen op box of domeinnaam listen (box: jellepie@pandora.be, domein: *@pandora.be)
    - *Blacklisting*: alles wat hierin voorkomt zal in een speciale folder terecht komen of rechtsreeks gewist worden.
      - Markeren om te wissen?
      - Gewoon direct wissen?
      - In aparte folder? (SpamCan || Quarantaine)
        - Deze settings moeten bepaald worden bij registratie of in config. Best in config. User hiermee niet lastig vallen bij registratie.
      - Null-reverse path: auto-blacklist?
      - Bounce? ~~(mss niet is zelfde als zegge spam me)~~
      - spamcop
    - *Whitelisting*: alles wat hierin zit zal zeker in de Inbox terecht komen.

    - White krijgt voorrang op Black.
    - ~~Global list <> Local list?: Nee, mss wille sommige mense wel nasty pr0n ;-)~~

- **Registration:**
  - Bij registration *beperkt aantal gegevens* vragen, nutteloos van hele boel data te vragen die nooit zal gebruikt worden. Eens alle data ingevoerd is zal getracht worden een connectie te maken adhv de gegeven data als dit niet lukt user de vraag stellen op accnt toch gemaakt moet worden. Sommige POP3 servers bieden geen toegang tot hosts die niet in hun ACL zitten. Bij registratie kan slechts 1 account gemaakt worden.
  - *Username / login*: unieke naam, zal gechecked worden met logins reeds aanwezig in dbase. (kan bvb een e-mail adres zijn evenals een gewone username, jellepie@pandora.be of gewoon jellepie, de user beslist hierover)

- *Password*: beperkt aantal chars (max. 20) MD5 of DES encryptie?
  - Ja: passwd opvragen wordt onmogelijk. Misschien door vraag die user bepaald maar zal stukken onveiliger zijn dan gewoon cleartext passwd.
  - Nee: cleartext passwd kan gemailed worden naar een reeds aangemaakte account met e-mail adres. Indien geen e-mail adres opgegevens too bad. Verplichting opgave? JA
- *POP3 Server*: FQDN of IP ( gethostname() && gethostbyaddr()) )
- *POP3 Secure*: SSL of noSSL (TLS)
- *POP3 User*: string
- *POP3 Pass*: string
- *E-mail adres*: voor de FROM header en voor passwd retrieval. Of probs kan zelfde zijn of ander, NEE zelfde

- **Adresboek**: on / off (global)
  - *On*: users kunnen een adresboek aanmaken en hierin contacten opslagen op dbase.
  - *Off*: users kunnen geen adresboek beheren onder hun account en dus geen contacten opslagen in dbase.
  - *Gaat zowiezo niet als registration op OFF staat.*
  - Contacten komen automatisch op whitelist.
    - Instellen door user in config. Standaard aan.
  - Bijgehouden data beperken tot min. alweer: het heeft geen nut 10-tallen zaken over een contact bij te houden, essentiële data only:
    - E-mail adres
    - Naam
    - Voornaam
    - Opmerkingen veld met persoonlijke zever (beperkt: 500 chars)
    - Fasty (naam die gebruikt wordt voor veelgebruikte contacten deze kan dan verschijnen in een box naast een nieuwe compose)

- **Adresboek import**: CSV of ~ XML ~
  - Vanuit OE, moet specifieke opmaak ebben
  - Contacten importeren vanuit een csv file. Voor xml is nog geen standaard.

- **Mailing list:**
  - Global beperking hierop?
    - Ja: n / nee: … / disable
  - Een groep users in een lijst (max 100 of meer maar dan moet auto. gesplitst worden)
    - List naam
    - Aantal contactnames met hun e-mail adres (links in dbase)

- **Compose e-mail:**
  - *From*: als meerdere accnts (standaard instellen in config)
  - *To*: List || één of meerdere contacts || beide
  - *Cc*: List || één of meerdere contacts || beide
  - *Bcc*: List || één of meerdere contacts || beide
  - *Request read receipt*
  - *Prioriteit*: hoog, normaal, laag (standaard normaal: instellen in config?)
  - *Attachs*: max size en aantal instellen op globale basis.
  - *Message*: HTML of Plaintext.

- **Message list: (specific folder)**
  - *Priority*: hoog, normaal, laag.
    - Dmv symbolen?
  - *Attachment*: aangeven dmv een symbool
  - *From*:
  - *Subject*:
  - *Received*: datum en uur van ontvangst
  - *Size*:
  - *Account*: als meerdere accounts (instellen in config)
  - *Sent*: datum dat de verzender het verstuurd heeft (instelle in config standaard off)
  - *To*: weergeven in popup (geel kaderke) (instellen in config: standaard off)
  - *Message sender box of domain markeren voor white of blacklist*
  - *Message markeren voor verwijdering*
  - *Message rule maken op basis van geslecteerd bericht … uitdokteren*
  - *Knoppen voorzien voor deze 'SpecOps'*
  - *Berichten die gelezen zijn laten zien !*

- **View message**: (geselecteerd uit list)
  - Wat kan je dan zien?
    - Priority:
    - Attachments (clickable om ze te bekijken)
    - From:
    - Subject:
    - Received:
    - Size:
    - Account
    - Sent
    - To
    - Message zelf
  - Wat kan je dan ermee doen?
    - Delete (from server)
      - mss beter delete overlaten aan overview?
    - B/W List it
      - Sender
    - Add sender to addressbook
    - Add all recipients to addressbook?? Naaah mss asser taaijm is
    - Message rule maken op basis van dit berich … uitdokteren
    - Forward (forward as attachment ook?)
      - Liever ni FW as attach omdat alle readers toch ni same format bezige
      - 'Nice' FW alles wordt gewist buiten laatste FROM
    - Reply → compose met ingevulde waarden? (op basis van msg)
      - 'Nice' reply: alles wordt gewist buiten subject
    - Reply to all (recipients and sender)
    - Print
    - Save ? (xml, pdf, doc, eml, … ?)
    - Volledige headers bekijken
    - Als er een read-receipt gevraagd is vragen of bevestigd moet worden (met delay ☺)

- **message rules / filters (/ highlighting)**
  - elk bericht wordt onderzocht en op basis van deze rules behandeld
  - IF { from, to, subject, cc, body, msg is prioritized, }
    { equals, (does not) contain(s) …, starts with, ends with }
    THEN
    { delete, put in folder, highlight }
  - Highlight: bold || italic || underline |& color
  - Stop processing more rules
    - Order of rules has importance !

- **compose new message**
  - *from*: als er meerdere accnts gedefinieerd zijn
  - *to, cc, bcc*: regelbaar via popup of zijbar
  - *message*: html of txt
  - *read receipt*
  - *priority*
  - *subject*
  - *attachment*

- **statistics (zeer beperkt)**
  - n° e-mail caught per filter (moet gesaved worden = dbase overhead)
  - n° e-mail blacklisted
    - domain (kan afgeleid worden moet ni apart gesaved worden)
    - box (idem voor domain)
    - whitelisted (idem voor 2 bovestaande)
  - n° contacts

- **signature**
  - 1 per account
  - Optie on / off
  - Beperkte lengte

- **Global ISP <> standalone**
  - **ISP**: POP3 server wordt globaal ingesteld en enkel users die een accnt hebben op die server kunnen POP3ye gebruiken
    - Mogelijkheid tot batching (lijst users ingeven in csv || xml en accnts worden aangemaakt)
  - **standalone**: POP3ye is toegankelijk voor alle users

- **Help FAQ**
  - Bestaat uit:
    - Categorie
      - Vraag
      - Antwoord

- **Local options**

- N° msgs per page
- Default sort order (date, from, size, …)
- Smileys: on / off
- Nice: on / off
- …

- **Smileys**
  - Op basis van bepaalde combis in mail (no A.I.)
  - Ook bij send???

- **Multi-language**
  - NL
  - EN
  - FR
  - GE
  - ES
  - Beperkt houden

- **Template / css**
  - Header
  - Footer
  - Css voor opmaak tss header en footer
  - Header en footer vaste linklokaties (abook, inbox, …)

# 3.1 Local PS vs. Global PS

## 3.1.1 Global POP3ye Settings (GPS)

How many filters can a user create: **n**
How many contacts can a user create: **n**
How many maillists can a user create: **n**
How many bw listings can a user create: **n**
How many POP3 accounts can a user create: **n**
Registration required for POP3ye use: **on / off**
POP3ye ISP mode: **on / off**
If **on**:
POP3ye domain: **string**
Null reverse-path auto blist: **on / off**
Blist default behaviour: **D / H (delete / highlight)**
Default stylesheet: **string**
Default theme: **string**
Default lang: **string**
Email suffix: **on / off**
If **on**:
Suffix: **string**
Addressbook import from csv,xml: **on / off**
DNS blisting: **on / off**

### 3.1.2 Local POP3ye Settings (LPS)

Zero Tolerance (ZT): **on / off**
SpamCop DNS blist checking: **on / off**
Signature: **string**
Account
    Name: **string**
    Server: **string**
    SSL: **yes / no**
    USER: **string**
    PASS: **string**
    Email: **string**
Default mail sort order: **string**
Bwlisting: **on / off**
Add a person I reply to to my abook: **on / off**
Add a person I reply to to my wlist: **on / off**
Null reverse-path blist: **D / H (delete / highlight)**
N° msgs per page: **n**
Nice on re:: **on / off**
Lang: **string**
Default priority: **string**
Default Ask for read receipt: **on / off**

## 3.2 AntiSpam

### 3.2.1 Filters

### 3.2.2 Black & White

### 3.2.3 DNS lookup

#### 3.2.3.1 SpamCop

If you want to use the list manually, or from custom software, you should instruct your system to do a dns query for the information. For example, if you want to check if 1.2.3.4 is on the blacklist, you might type this at the command-line:

nslookup 4.3.2.1.bl.spamcop.net

If you get back an IP address (typically 127.0.0.2), then the IP you asked about is listed. If you get back a non-existant message, then the IP you asked about is not listed.

"nslookup" is just the most common method for looking up a hostname. Your system may have another name for it. Other common names are "host" and "dig".

# 4 e-mail protocols aanspreken via PHP

## 4.1 php_imap

**IMAP, POP3 AND NNTP FUNCTIONS**

| Function | Description | Returns |
|---|---|---|
| `imap_8bit(string)` | Converts an 8-bit string to a quoted-printable string. | String |
| `imap_alerts()` | Returns all IMAP alert messages for this page request. | Array |
| `imap_append(imap_stream, mbox, message [, flags])` | Appends a string message to a mailbox. | Integer |

**IMAP, POP3 AND NNTP FUNCTIONS (CONTINUED)**

| Function | Description | Returns |
|---|---|---|
| `imap_base64(text)` | Decodes base64 encoded text. | String |
| `imap_binary(string)` | Converts an 8-bit string to a base64 string. | String |
| `imap_body(imap_stream, msg_number [, flags])` | Returns a message body. | String |
| `imap_check(imap_stream)` | Checks the current mailbox. | Object |
| `imap_clearflag_full(stream, sequence, flag, options)` | Clears message flags. | String |
| `imap_close(imap_stream [, flags])` | Closes an IMAP stream. | Integer |
| `imap_createmailbox (imap_stream, mbox)` | Creates a new mailbox. | Integer |
| `imap_delete(imap_stream, msg_number [, flags])` | Marks a message for deletion from the current mailbox. | Integer |
| `imap_deletemailbox (imap_stream, mbox)` | Deletes a mailbox. | Integer |
| `imap_errors()` | Returns all IMAP errors for this page request. | Array |
| `imap_expunge(imap_stream)` | Deletes all messages marked for deletion. | Integer |
| `imap_fetch_overview (imap_stream, sequence [, flags])` | Returns an overview of header information for a message. | Array |
| `imap_fetchbody(imap_stream, msg_number, part_number [, flags])` | Fetches a section of the body of a message. | String |
| `imap_fetchheader(imap_stream, msg_number, flags)` | Returns the header for a message. | String |
| `imap_fetchstructure (imap_stream, msg_number [, flags])` | Returns the structure of a message. | Object |
| `imap_get_quota(imap_stream, quota_root)` | Returns the quota level settings and usage statistics for a mailbox. | Array |
| `imap_getmailboxes (imap_stream, ref, pattern)` | Returns detailed information about each mailbox. | Array |
| `imap_getsubscribed (imap_stream, ref, pattern)` | Returns a list of all the subscribed mailboxes. | Array |
| `imap_header(imap_stream, msg_number [, fromlength [, subjectlength [, defaulthost]]])` | Returns the header of a message. | Object |
| `imap_headerinfo(imap_stream, msg_number [, fromlength [, subjectlength [, defaulthost]]])` | Returns the header of a message. | Object |

## IMAP, POP3 AND NNTP FUNCTIONS (CONTINUED)

| Function | Description | Returns |
|---|---|---|
| `imap_headers(imap_stream)` | Returns the headers for all messages in a mailbox. | Array |
| `imap_last_error()` | Returns the last IMAP error for this page request. | String |
| `imap_listmailbox (imap_stream, ref, pattern)` | Returns a list of mailboxes. | Array |
| `imap_listsubscribed (imap_stream, ref, pattern)` | Returns a list of all the subscribed mailboxes. | Array |
| `imap_mail(to, subject, message [,additional_headers [, cc [, bcc [, rpath]]]])` | Sends an e-mail message. | String |
| `imap_mail_compose(envelope, body)` | Creates a MIME message using the provided envelope and body. | String |
| `imap_mail_copy(imap_stream, msglist, mbox [, flags])` | Copies messages to a mailbox. | Integer |
| `imap_mail_move(imap_stream, msglist, mbox [, flags])` | Moves messages to a mailbox. | Integer |
| `imap_mailboxmsginfo (imap_stream)` | Returns information about the current mailbox. | Object |
| `imap_mime_header_decode(text)` | Decodes MIME header elements. | Array |
| `imap_msgno(imap_stream, uid)` | Returns the message sequence number for a UID. | Integer |
| `imap_num_msg(imap_stream)` | Returns the number of messages in the current mailbox. | Integer |
| `imap_num_recent(imap_stream)` | Returns the number of recent messages in the current mailbox. | Integer |
| `imap_open(mbox, username, password [, flags])` | Opens an IMAP stream to a mailbox. | Integer |
| `imap_ping(imap_stream)` | Checks if an IMAP stream is still active. | Integer |
| `imap_qprint(string)` | Converts a quoted-printable string to an 8-bit string. | String |
| `imap_renamemailbox (imap_stream, old_mbox, new_mbox)` | Renames a mailbox. | Integer |
| `imap_reopen(imap_stream, mbox [, flags])` | Re-opens an IMAP stream to a new mailbox. | Integer |
| `imap_rfc822_parse_adrlist (address, default_host)` | Parses an address string. | Array |
| `imap_rfc822_parse_headers (headers [, defaulthost])` | Parses mail headers from a string. | Object |
| `imap_rfc822_write_address (mbox, host, personal)` | Returns a formatted e-mail address from the provided mailbox, host and personal information. | String |
| `imap_scanmailbox(imap_stream, ref, pattern, content)` | Searches the names of mailboxes for a specified string. | Array |
| `imap_search(imap_stream, criteria, flags)` | Returns messages matching the provided criteria. | Array |
| `imap_set_quota(imap_stream, quota_root, quota_limit)` | Sets a quota for a mailbox. | Integer |
| `imap_setflag_full(stream, sequence, flag, options)` | Sets flags on messages. | String |

**IMAP, POP3 AND NNTP FUNCTIONS (CONTINUED)**

| Function | Description | Returns |
|---|---|---|
| `imap_sort(stream, criteria, reverse, options)` | Sorts message headers. | Array |
| `imap_status(imap_stream, mbox, options)` | Returns status information on a mailbox other than the current mailbox. | Object |
| `imap_subscribe(imap_stream, mbox)` | Subscribes to a mailbox. | Integer |
| `imap_uid(imap_stream, msg_number)` | Returns the UID for a message sequence number. | Integer |
| `imap_undelete(imap_stream, msg_number)` | Unmarks a message for deletion. | Integer |
| `imap_unsubscribe (imap_stream, mbox)` | Unsubscribes from a mailbox. | Integer |
| `imap_utf7_decode(text)` | Decodes a modified UTF-7 encoded string. | String |
| `imap_utf7_encode(data)` | Converts 8-bit data to modified UTF-7 text. | String |
| `imap_utf8(text)` | Converts text to UTF8. | String |

**MAIL FUNCTIONS**

| Function | Description | Returns |
|---|---|---|
| `ezmlm_hash(addr)` | Calculates the hash value required by EZMLM mailing lists. | Integer |
| `mail(to, subject, message [, additional_headers [, additional_parameters]])` | Sends mail. | Boolean |

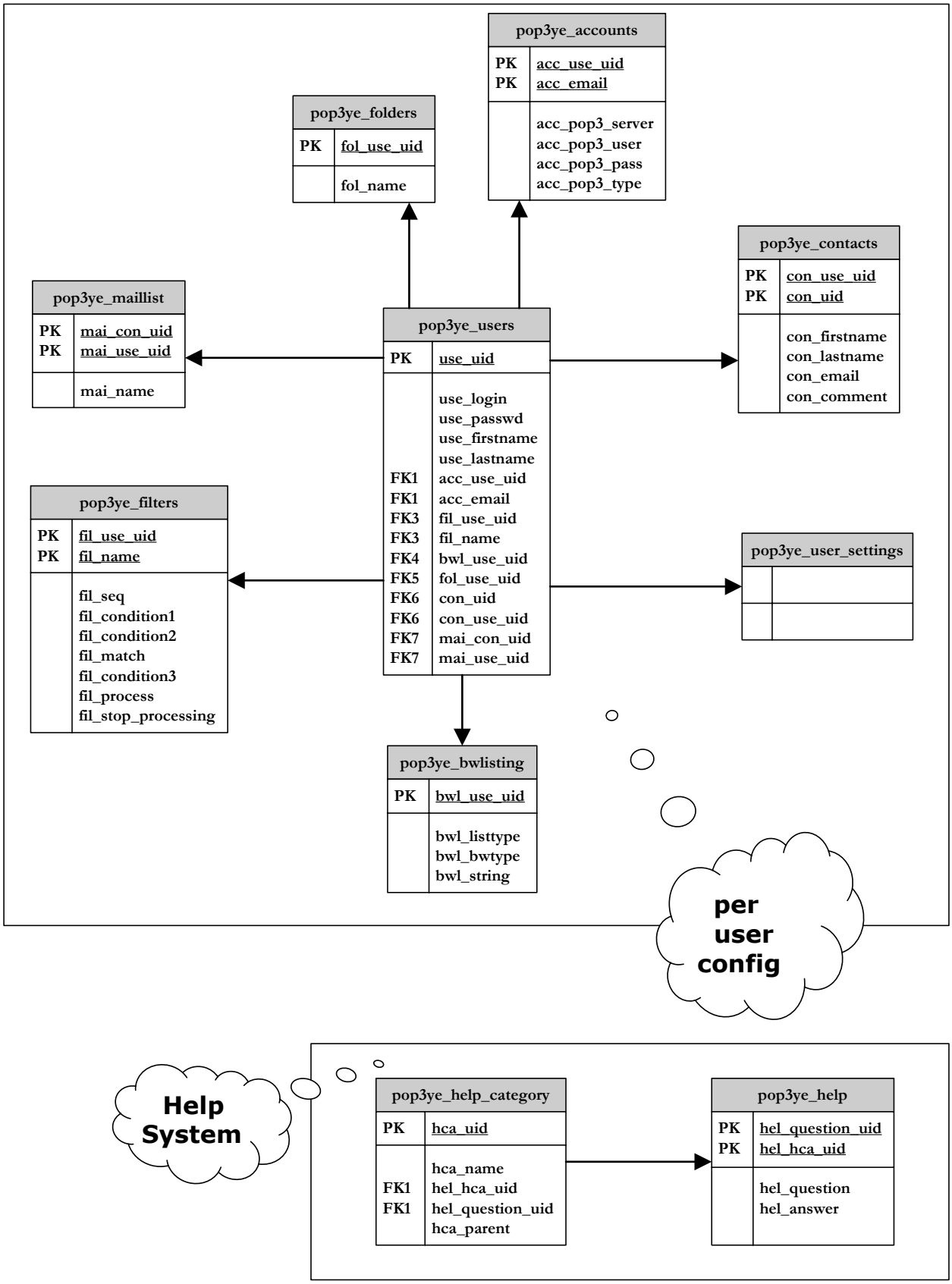
# 4.2 mail()

http://be2.php.net/manual/nl/function.mail.php

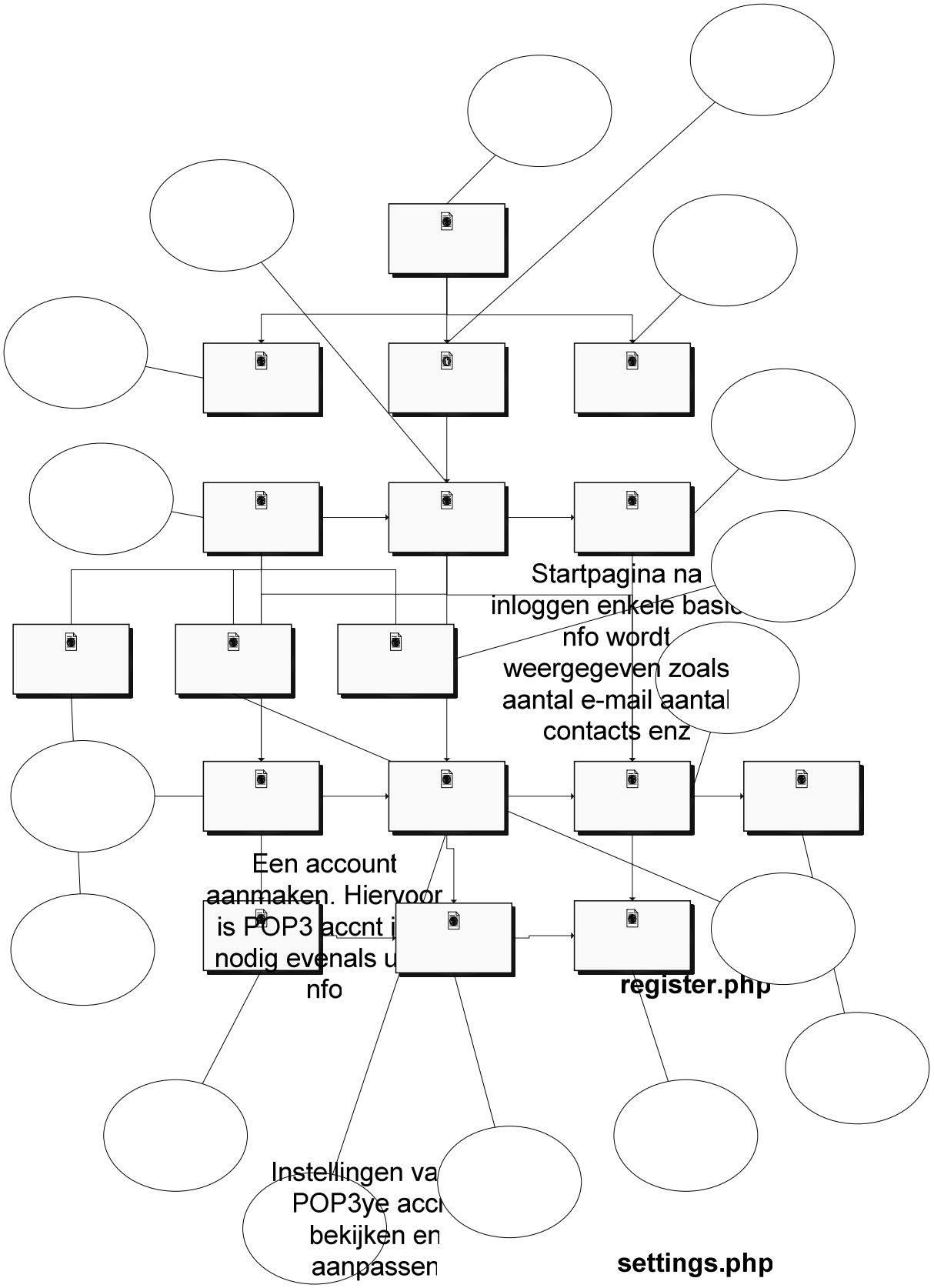
**MAIL FUNCTIONS**

| Function | Description | Returns |
|---|---|---|
| `ezmlm_hash(addr)` | Calculates the hash value required by EZMLM mailing lists. | Integer |
| `mail(to, subject, message [, additional_headers [, additional_parameters]])` | Sends mail. | Boolean |

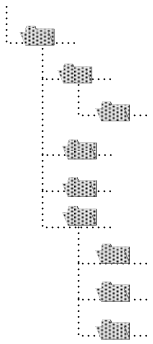
# 5 Database design



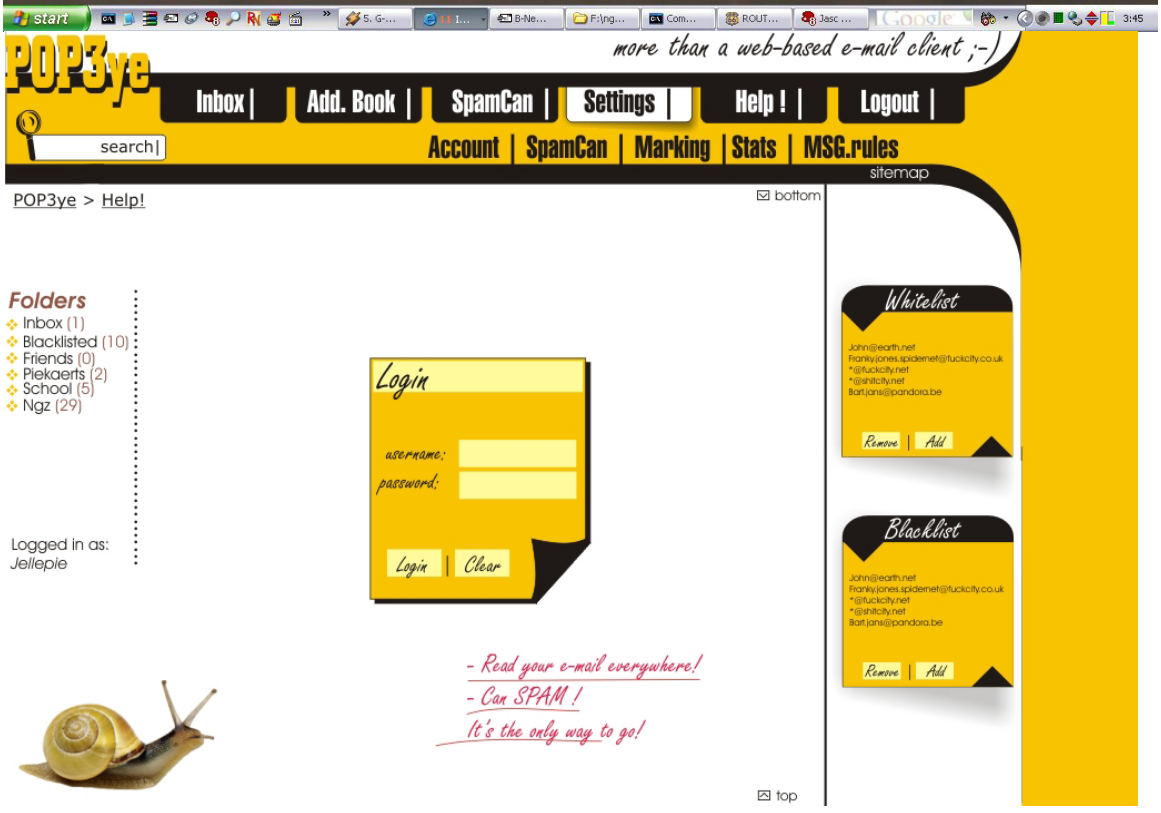
# 6 File en directory structuur



| | |
|---|---|
| \ | *alle gewone .php files* |
| \includes | *include files van verschillende types* |
| \js | **.js JavaScript includes* |
| \php | **.inc.php PHP includes* |
| \css | **.css Stylesheets* |
| \grafx | *graphics & template design elements* |
| \lang-nl | *in verschillende talen indien nodig* |
| \lang-en | |
| \lang-fr | |
| \lang-es | |
| \ ... | |
| \lang | *taalbestanden* |
| \admin | *administratie module (php files)* |



# 7 Layout

# 8 PHP5

## 8.1 Nieuw in PHP5

Changes in PHP5 / Zend Engine 2.0
http://www.php.net/zend-engine-2.php

PHP5 Changelog:
http://www.php.net/ChangeLog-5.php#5.0.0b4

Introduction to PHP5:
http://www.phpbuilder.com/columns/argerich20030411.php3

PHP5: Coming to a webserver near you soon:
http://www.sitepoint.com/print/1192

Enkele:

- Better OO support
- Private
- Public
- Protected
- Abstract
- Better xml support
- error handling
- __construct()
- __destruct()
- __call()
- __clone()
- ….

## 8.2 Configuratie

Php.ini aanpassingen waaronder hide all errors

## 8.3 Installatie

Install Apache 2 (apache_2.0.48-win32-x86-no_ssl.msi)

extract php naar c:\php (php5-win32-200402071130.zip)

copy php5ts.dll & php5apache2.dll naar folder waar apache.exe zit

---

configure httpd.conf

```
LoadModule php5_module c:/php/php5apache2.dll
        #AddModule mod_php5.c
AddType application/x-httpd-php .php
```

configure php.ini

```
extension_dir = "c:/php/ext/"
extension = php_imap.dll
```

# 9 Test –en ontwikkelingsomgeving

## 9.1 Serversoftware

### 9.1.1 Apache

### 9.1.2 mySQL

**MySQL Functions**

| Function | Description | Returns |
|---|---|---|
| `mysql_affected_rows(link_id)` | Returns the number of rows affected by the last query. | Integer |
| `mysql_change_user(user, password [, database [, link_id]])` | Changes the logged in user of the connection. | Integer |
| `mysql_close([link_id])` | Closes the connection to a MySQL database. | Integer |
| `mysql_connect([hostname [:port] [:/path/to/socket] [,username [,password]]])` | Opens a connection to a MySQL database. | Integer |
| `mysql_create_db(database [, link_id])` | Creates a MySQL database. | Integer |
| `mysql_data_seek(result_id, row_number)` | Moves to a specified row in a result set. | Integer |
| `mysql_db_name(result_id, row [, field])` | Returns result data. | Integer |
| `mysql_db_query(database, query [, link_id])` | Executes a query. | Integer |
| `mysql_drop_db(database_name [, link_id])` | Deletes a MySQL database. | Integer |
| `mysql_errno([link_id])` | Returns the error number from the last MySQL operation. | Integer |

| Function | Description | Returns |
|---|---|---|
| mysql_error([link_id]) | Returns the error message from the last MySQL operation. | String |
| mysql_fetch_array(result_id [, result_type]) | Returns the next row as an array. | Array |
| mysql_fetch_assoc(result_id) | Returns a row as an associative array. | Array |
| mysql_fetch_field(result_id [, field_offset]) | Returns field information. | Object |
| mysql_fetch_lengths(result_id) | Returns the length of each field in a result set. | Array |
| mysql_fetch_object(result_id [, result_type]) | Returns the next row as an object. | Object |
| mysql_fetch_row(result_id) | Returns the next row as an enumerated array. | Array |
| mysql_field_flags(result_id, field) | Returns the flags for a field in a result set. | String |
| mysql_field_len(result_id, field) | Returns the length of a field in a result set. | Integer |
| mysql_field_name(result_id, field) | Returns the name of a field in a result set. | String |
| mysql_field_seek(result_id, field_offset) | Moves to a specified field offset. | Integer |
| mysql_field_table(result_id, field) | Returns the name of the table a field belongs to. | String |
| mysql_field_type(result_id, field) | Returns the type of a field in a result set. | String |
| mysql_free_result(result_id) | Frees the memory used by a result set. | Integer |
| mysql_insert_id([link_id]) | Returns the ID generated from the last INSERT operation. | Integer |
| mysql_list_dbs([link_id]) | Lists the databases on a MySQL server. | Integer |
| mysql_list_fields(database, table [, link_id]) | Lists the fields in a table. | Integer |
| mysql_list_tables(database) | Lists the tables in a MySQL database. | Integer |
| mysql_num_fields(result_id) | Returns the number of fields in a result set. | Integer |
| mysql_num_rows(result_id) | Returns the number of rows in a result set. | Integer |
| mysql_pconnect([hostname [:port] [:/path/to/socket] [, username [, password]]]) | Opens a persistent connection to a MySQL database. | Integer |
| mysql_query(query [, link_id]) | Executes a query. | Integer |
| mysql_result(result_id, row [, field]) | Returns the contents of a cell in a result set. | Mixed |
| mysql_select_db(database [, link_id]) | Makes a MySQL database the current database. | Integer |
| mysql_tablename(result_id, index) | Returns the name of the table for a field. | String |

### 9.1.3 POP3 en SMTP onafhankelijk ( bvb qpopper )

## 9.2 Ontwikkelingssoftware