Requirements Specification Document for Database-based email system

Group No. 12
Rukma Prabhu Verlekar 04305601
Neela Sawant 04305811
Meghana Kshirsagar 04305907

September 10, 2004

Contents

1 Introduction

1.1 Scope of Project

We are planning to build a Database-based email-system which will have the following functionalities:

1. An SMTP Server
2. An IMAP Server
3. An LDAP based authentication system.
4. A database at the back-end of the mail-server.
5. Server-side Mail-filtering
6. A client-plugin for Squirrel-mail.

1.2 Motivation

Many of the existing email-systems have some short-comings in terms of the functionalities that they offer. We would like to implement a set of such most-sought-after features that users want to see in their email-system.

1.3 Overview of System

A database-based email system has a DBMS at its backend to store all user-data, including settings, address book, calendar, authentication, filters, Spam settings and more. Apart from the basic functionality offered by a DBMS, this approach has the following advantages:

- Faster retrieval:
  Mail and other data can be retrieved faster when compared to traditional file-system-based email systems.
- Savings in Storage:
  Mails sent to mailing-lists needn't be replicated for every user on the list. They are stored only once in the file system.
- Faster Search:
  Searching mails becomes more efficient with the use of database-queries.

2 General Description

2.1 Scope of this document

This document elaborates the features of the above described system that our group is going to implement in this project. These are:

1. An SMTP Server
2. An LDAP based authentication system.
3. A database at the back-end of the mail-server.

2.2 Functional Description

2.2.1 SMTP Server

- The SMTP server:
  An SMTP server is concerned with the reliable and efficient transfer of mail. It delivers the mail sent by the user to the destination mail-box.

- Mail Relay:
  We are also going to implement the Mail Relay feature in our SMTP server which has the capability of relaying mail across transport service environments. An SMTP server may be either the ultimate destination or an intermediate "relay" for a message. That is, it may assume the role of an SMTP client after receiving the message if it doesnt have a mailbox for that particular user. In such cases the SMTP server obtains the remote ID using MX records in the DNS and gets the address of the remote SMTP server.

2.2.2 Mailing Lists

Mailing list is the solution for sending newsletters, distributing messages to different mailing lists, sending notifications to the users, as well as for sending individual messages. It avoids redundant storage of group messages.
Mailing list features:

1. Simultaneously deliver many messages.
2. Single copy of the sent message needs to be stored.
3. Eliminates bounced e-mails.
4. Logging of bad e-mail addresses so that the lists can be filtered.

2.2.3 The Back-end Database

We are going to use the Postgress DBMS at the back-end of our email server. We propose to use Stored Procedures to make the database access more efficient.
The email will be broken into the following parts before it is stored in the database.

- Header
- Body (includes the attachment)

Since the attachments are MIME encoded, they are stored as a part of the body. The database contains for each mail-header part, the location of the body part that is stored as a blob on the file system.

2.2.4 LDAP-based Authentication

We will use LDAP-based authentication instead of getting user information from the back-end database. LDAP uses a different model of data: a tree model called the Directory Information Tree (DIT) and we can imagine each entry as either a fork or a leaf on the tree. In a nutshell, LDAP provides central management of access, authentication, and authorization. It's easily customizable and can:

- Centralize user and group management
- Centralize information stores
- Set security and access control
- Securely delegate read and modification authority
- Serve almost any platform
- Scale efficiently

2.2.5 Importing Mail

Our email-system will be capable of importing the INBOX for a user from other domains. This will provide the facility that given an IMAP-server address, it will pull all its mails to the user's system.

3 Additional features proposed

3.1 Sending HTML mail

Users will be able to send HTML mails, apart from plain-text. HTML-formatted mail, allows you to style the text (bold, italics etc.), add pictures etc. This will require the support for the corresponding MIME format.

4 Conclusion

A Database-based email system is a considerable improvement over traditional file-based email systems in that it provides a better management of the email-data and faster search and retrieval. Also the benefits of using a DBMS will allow us to provide more flexibility and options in querying the data.