Lemonade Notifications Architecture

Abstract

   Notification and filtering mechanisms can make email more enjoyable
   on mobile and other constrained devices (such as those with limited
   screen sizes, memory, data transfer rates, etc.). Notifications make
   the client aware of significant events (such as the arrival of new
   mail) so it can react (such as by fetching interesting mail
   immediately). Filtering reduces the visible mail to a set of
   messages that meet some criteria for "interesting". This
   functionality is included in the goals of the Lemonade (Enhancements
   to Internet email to Support Diverse Service Environments) Working
   Group.

   This document also discusses the use of server-to-server
   notifications, and how server to server notifications fit into an
   architecture that provides server to client notifications.

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

outside the IETF Standards Process, and derivative works of it may
not be created outside the IETF Standards Process, except to format
it for publication as an RFC or to translate it into languages other
than English.

Table of Contents

1. Introduction

   The Lemonade work [LEMONADE-PROFILE] identified a need to provide
   notification and filtering mechanisms for use with IMAP [IMAP].

   In addition, external groups that make use of IETF work also
   expressed such requirements (see, for example, [OMA-LEMONADE-ARCH];
   Open Mobile Alliance (OMA) requirements for within-IMAP ("inband")
   and out-of-IMAP ("outband") server to client notifications are listed
   in [OMA-ME-RD]).

1.1. Conventions Used in This Document

   Within this document, the terms "Lemonade Profile" and "Lemonade"
   generally refer to the revised Lemonade Profile document, RFC 5550
   [LEMONADE-PROFILE].

2. Notifications Logical Architecture and LEMONADE Profile

   The target logical architecture for the LEMONADE Profile is described
   in the revised Lemonade Profile document [LEMONADE-PROFILE].

   Figure 1 illustrates how notification and filtering fit in the
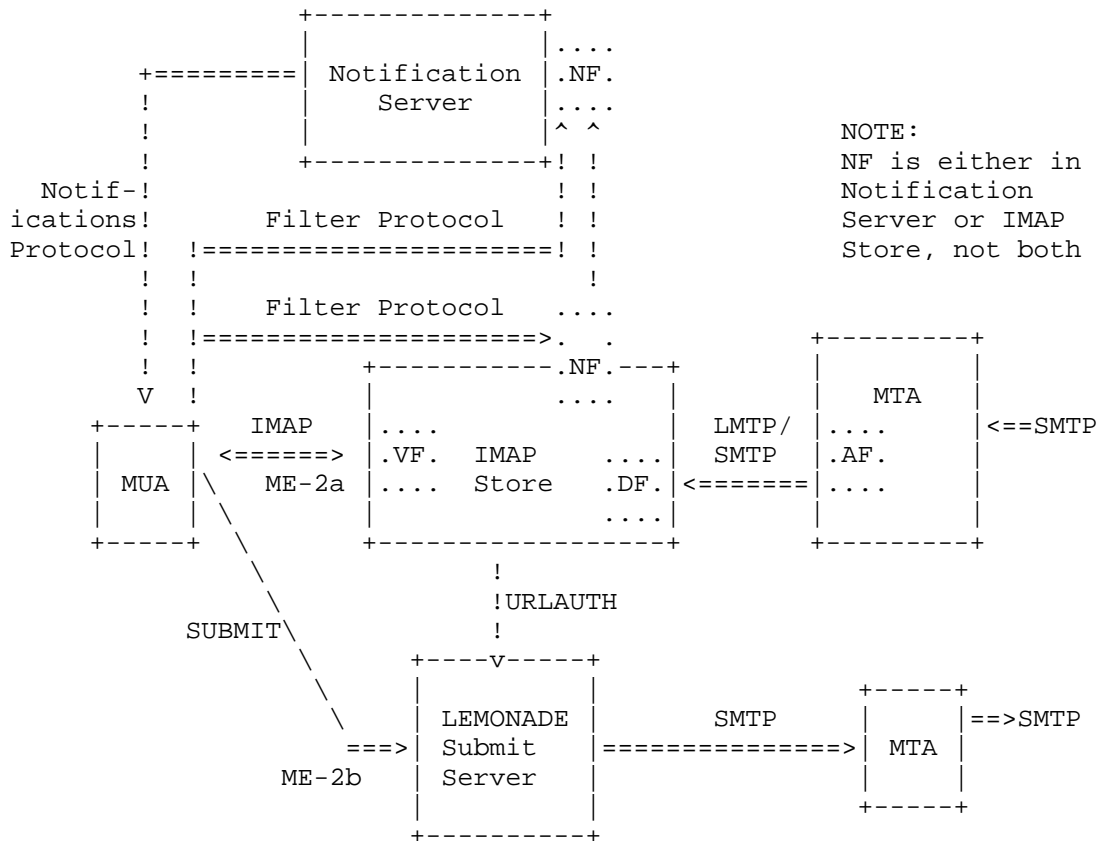   context of Lemonade.

```
                       +--------------+
                       |              |....
             +=========| Notification |.NF.
             !         |    Server    |....
             !         |              |^ ^
             !         +--------------+! !
     Notif-!                          ! !
     ications!        Filter Protocol ! !
     Protocol!   !=====================! !
             ! !                         !
             ! !   Filter Protocol  ....
             ! !=====================>.  .
             ! !         +-----------.NF.---+            +---------+
             V !         |           ....   |            |         |
       +-----+   IMAP    |....              |  LMTP/  |  |   MTA   |
       |     | <======>  |.VF.  IMAP    ....|  SMTP   |....     |<==SMTP
       | MUA |\  ME-2a   |....  Store   .DF.|<=======|....     |
       |     | \         |             ....|         |         |
       +-----+  \        +-----------------+         +---------+
                 \                 !
                  \            !URLAUTH
          SUBMIT\            !
                 \      +----v-----+
                  \     |          |            +-----+
                   \    | LEMONADE |    SMTP     |     |==>SMTP
              ===>| Submit   |===============>| MTA |
          ME-2b     | Server   |            |     |
                    |          |            +-----+
                    +----------+
```

Figure 1: Filtering Mechanism Defined in
Lemonade Profile Architecture

NOTE:
NF is either in
Notification
Server or IMAP
Store, not both

In Figure 1, four categories of filters are defined:

1. AF: Administrative Filters: Created and maintained by mail
   administration. AF are typically not configured by the user and
   are used to apply policies, content filtering, virus protection,
   spam filtering, etc.

2. DF: Deposit Filters: Executed on deposit of new mail. Can be
   defined as Sieve filters [SIEVE].

3. VF: View Filters: Define which messages are important to a
   client. May be implemented as pseudo-virtual mailboxes [CONTEXT].
   Clients may use this to restrict which messages they synchronize.

   4. NF: Notification Filters: Determine when out-of-IMAP ("outband")
      notifications are sent to the client. These filters can be
      implemented either in the message store or in a separate
      notifications engine.

   Note that when implementing DF or NF using Sieve, the 'enotify'
   [SIEVE-NOTIFY] and likely the 'variables' [SIEVE-VARIABLES] Sieve
   extensions might be needed.

   The filters are manageable by the client as follows:

   * NF and DF: When internal to the mail store, these are typically
     implemented using Sieve; hence, a Sieve management protocol is used
     for client modifications. See [MANAGE-SIEVE] for more information.
     Per-mailbox notifications might be implemented using a combination
     of a primary Sieve script for notifications on delivery into a
     mailbox (e.g., FILEINTO) and a per-mailbox Sieve script such as
     [IMAP-SIEVE] for transfers into a mailbox. When the NF is within a
     notification server, it is out of scope of Lemonade.

   * VF: via pseudo-virtual mailboxes as defined in [CONTEXT].

   In Figure 1, the NF are shown both as part of the mail store (for
   example, using Sieve) and as an external notification server. Either
   approach can be used.

3. Event-Based Synchronization

```
   +----------------+       +--------------+       +------------+
   |    COMPLETE    | (VF)  |     VIEW     |  (NF) |    PUSH    |
   |   REPOSITORY   | View  |  REPOSITORY  |Notification| REPOSITORY |
   |                |Filters|              | Filters |            |
   |    all email   |       |  email to be |       |  important  |
   | in the account |=======|synched by the|=====<?>====| email /     |
   |                |       | mobile client|       | | events      |
   |                |       |   (CONTEXT)  |       | |            |
   +----------------+       +--------------+       |  +------------+
                                                   |         |
                                               IDLE /        |
                                               NOTIFY    Out-of-IMAP
                                                   |    Notifications
                                                   |         |
                                                   V         V
```
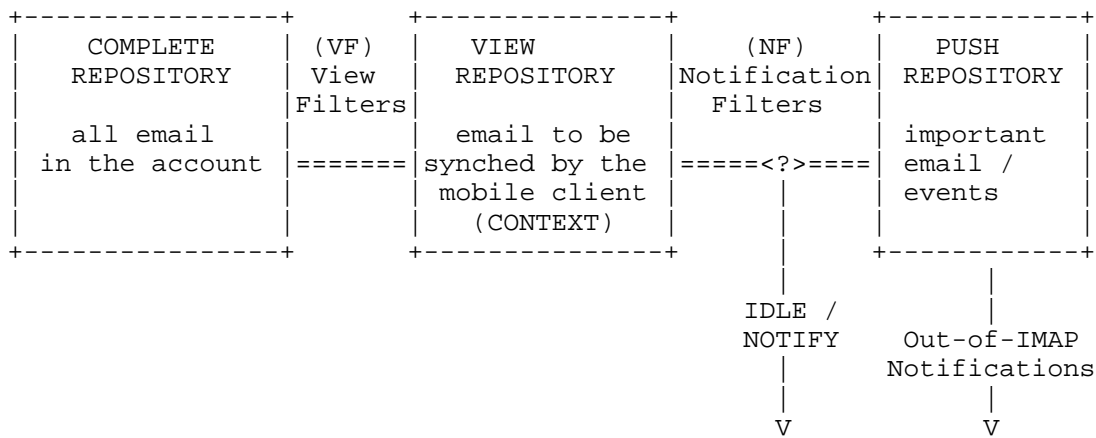
            Figure 2: Filters and Repositories

For in-IMAP ("inband") notifications, the Mail User Agent (MUA)
(client) issues IDLE [IDLE], or the successor extension command
NOTIFY [NOTIFY]; the LEMONADE IMAP server sends notifications as
unsolicited responses to the client.

Out-of-IMAP ("outband") notifications are messages sent to the user
or client not through IMAP. When directed at the user, they are
human-consumable and intended to alert the user. When directed at
the client, they are machine-consumable and may be acted upon by the
receiver in various ways, for example, fetching data from the mail
store or resynchronizing one or more mailboxes, updating internal
state information, and alerting the user.

4. Push Email

A good user experience of "push email" requires that when
"interesting" events occur in the mail store, the client is informed
so that it can connect and resynchronize. The Lemonade Profile
[LEMONADE-PROFILE] contains more information, especially in Section
5.4.2, titled "External Notifications".

5. Server-to-Server Notifications Rationale

With server-to-server notifications, a mail system generates event
notifications. These notifications describe mailbox state change
events such as arrival of a new message, mailbox full, and so forth.

See [MSGEVENT] for a list of such events.

These state change notifications are sent to a notification system,
which may generate alerts or notifications for delivery to one or
more clients or the user.

Server-to-server notifications allow the mail system to generate end
user or client notifications without needing to keep track of
notification settings for users or clients; the notification system
maintains notification preferences for clients and users.

Using server-to-server notifications, the mail system can provide the
end user with a unified notification experience (the same look and
feel for accounts at all messaging systems, such as email and
voicemail), while allowing smooth integration of additional messaging
systems.

5.1. Notifications Discussion

   The POP3 and IMAP4 Internet mail protocols allow mail clients to
   access and manipulate electronic mail messages on mail systems. By
   definition and scope, these protocols do not provide off-line methods
   to notify an end user when the mailbox state changes. Nor does
   either protocol define a way to aggregate the status within the end
   user's various mailboxes.

   The desire for this functionality is obvious. For example, from the
   very early days of electronic mail, various notifications mechanisms
   have been used, including login shell checks, and simple hacks such
   as [BIFF].

   To provide an end user with unified notifications and one centralized
   Message-Waiting Indicator (MWI), notification mechanisms are needed
   that aggregate the information of all the events occurring on the end
   user's different messaging systems.

   Server-to-server notifications allow the messaging system to send
   state change events to the notification system when something happens
   in or to an end user's mailbox.

   Notification systems can be broadly grouped into three general
   architectures: external smart clients, intrinsic notification, and
   separate notification mechanisms.

   External smart clients are agents independent of the mail system that
   periodically check mailbox state (or receive notifications, for
   example, via IMAP IDLE) and inform the user or the user's mail
   client. Many such systems have been used over the years, including
   login shells that check the user's mail spool, laptop/desktop tiny
   clients that periodically poll the user's mail servers, etc.

   Intrinsic notification is any facility within a mail system that
   generates notifications, for example, the server component of [BIFF],
   or, for more modern systems, the recent Sieve extensions for
   notifications [SIEVE-NOTIFY].

   Separate notification systems decouple the state change event
   notification from the end user or client notification, allowing a
   mail system to do the former, and specialized systems (such as those
   that handle presence) to be responsible for the latter. This
   separation is architecturally cleaner, since the mail system only
   needs to support one additional protocol (for communication to the
   notification system) instead of multiple notification delivery
   protocols, and does not need to keep track of which clients and which
   users are interested in which events. It also allows notifications

   to be generated for any service, not just electronic mail. However,
   it requires a new service (the notification system) and the mail
   system needs to support an additional protocol (to communicate with
   the notification system).

   In addition to any external notification mechanisms, Sieve can be
   used for notifications [SIEVE-NOTIFY]. Since many mail systems
   already provide Sieve support, this can be a fairly easy and quick
   deployment option to provide a useful form of notifications.

5.2. Server-to-Server Notifications Scope

   For the purposes of the Lemonade work, the scope of server-to-server
   notifications is limited to communications between the mail system
   and the notification system (the third architectural type described
   in Section 5.1). Communication between the notification system and
   the end user or devices (which might use SMS, WAP Push, instant
   messaging, etc.) is out of scope. Likewise, the scope generally
   presumes a security relationship between the mail system and the
   notification system. Thus, the security relationship then becomes
   the responsibility of the notification system. However, the
   specifics of security, trust relationships, and related issues depend
   on the specifics of both server-to-server notifications and
   notification systems.

   Figure 3 shows the context of server-to-server notifications; only
   the left side is in scope for Lemonade:

```
                 +--------+                            +--------+
    New          |        |_                           |  SMS   |
  Message        |  Mail  | \                          |Gateway |
  ------->       |Server 1|  \                     __  |        |
                 +--------+   \                    /   +--------+
                    ^  \                          /
                    |   \                        /  ^
                    |    \   +-------------+     /   |    +--------+
                 +--------+  |             |    / /  |    |  MWI   |
    Read         | Voice  |  |  -| Notification |/    |    |Gateway |
  Message        |  Mail  |  |   | Server      |     |    |        |
  ------->       | Server |  | ^  __|         |\  ^  |    +--------+
                 +--------+  | | / |(out of scope)| \ |  |
                             | |/  |             |   \|  |
                             | / ^ +-------------+ ^  \| |
                             |/| |                \ | |\|
                 +--------+  /| |                 \| | |\ +--------+
  Mailbox        |        | /| | |                 | | | \| WAP    |
  Full           |  Mail  |/ | | |              ^ \ | | |\| Push   |
  ------->       |Server 2| / | | |              | |\| | \|Gateway |
                 +--------+ | | | |              | | | | \+--------+
                            | | | |              | | | |\|
                            | | | |              | | | | \
                            | | | |              | | | |  \ +--------+
                            | | | |              | | | |   \| IM     |
                            | | | |              | | | |   \|Gateway |
                            | | | |              | | | |    |        |
                            | | | |              | | | |    +--------+
                            | | | |              | | | |
                            | | | |              | | | |
                          Server-to-                OTHER
                            Server               PROTOCOLS
                          Notifications        (out of scope)
                          (in scope)
```
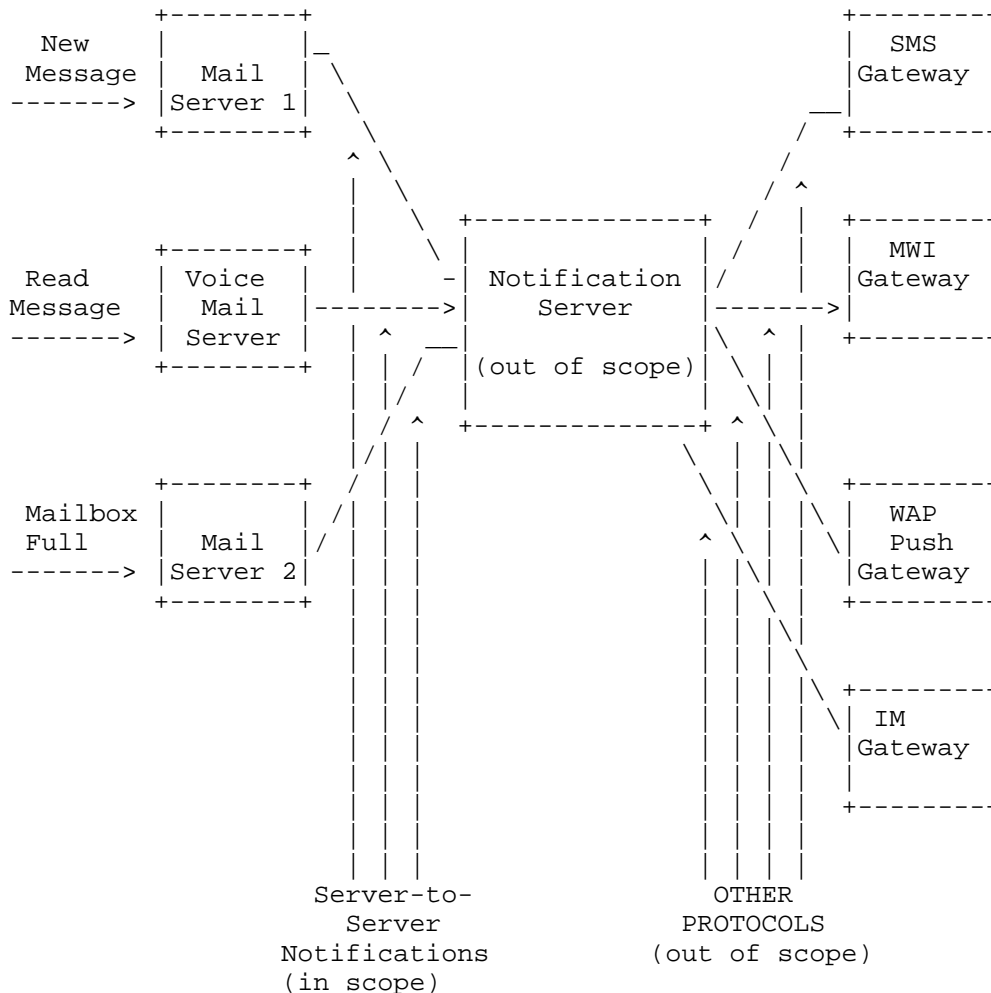
                    Figure 3: Scope of Server-to-Server Notifications

5.3. Basic Operation

   The mail system sends state change event notifications to the
   notification system (which in turn might notify a client or end user)
   for events that occur in the end user's mailboxes. Each such
   notification, referring to a single mailbox event, is called a state
   change event.

The state change event contains data regarding the mailbox event that
has occurred. The state change event describes the change, but
normally does not specify how or if the end user or client is
notified; this allows the end user and client notification
preferences to be maintained only within the notification server.

From the Lemonade viewpoint, out-of-IMAP (outband) notifications are
usually desired only when the client is not connected to the IMAP
server (since inband notifications are used when there is an IMAP
connection). Thus, it is helpful for the mail system to be able to
inform the notification system when the user logs in or out, and
which client is used (when this information is available).

When Sieve is used, the Sieve engine might have access to this
information.

A message is generated by the message store as a result of a state
change event. This message may be delivered to the end user, a
client, or to an external notification server that might deliver an
equivalent message to the user or to a client.

Within the context of the Lemonade Profile (Figure 1), the event is
filtered by NF. That is, the Notification Filters logically
determine which state change events cause notification to the user or
client.

Notifications allow for a rich end user experience. This might
include conveying mailbox status, new message attributes, etc., to
the user or client independent of the client's connection to the mail
store.

Notifications also allow for different Message Waiting Indicator
(MWI) behaviors (e.g., turn MWI indication off after all the messages
in all the end user's mailboxes have been read, should such an
unlikely thing occur in the real world).

The payload of a notification might include a URL referring to the
message that caused the event, possibly using URLAUTH [URLAUTH].

As state change events occur in the mail store, they are filtered,
which is to say matched against client or user preferences. As a
result, a notification may or may not be generated for delivery to
the user or client.

In the most general case, the mail system sends bulk state change
events to an external notification server, and it is the notification
server that filters the events by matching against the user's or
client's preferences.

   In the most mail-specific case, the mail system performs the
   filtering itself, for example, using Sieve.

5.4. Event Order

   For the Lemonade Profile, the event order is generally not important.
   By including information such as the modification sequence identifier
   (called a modseq or mod-sequence) [CONDSTORE] in notifications, the
   receiving client can quickly and easily determine if it has already
   processed the triggering event (for example, if a notification
   arrives out of order, or if the client has resynchronized since the
   event was generated).

   For generic server-to-server notifications, the order is likely to
   matter and the mail system needs to provide notifications to the
   notification system in the order that they occur.

5.5. Reliability

   For the Lemonade Profile, lost or delayed notifications to the client
   are tolerated. A client can resynchronize its state (including that
   reported by any missing events) when it next connects to the server.

   For generic server-to-server notifications, it is assumed that the
   data in a state change event is important, and therefore a high level
   of reliability is needed between the mail system and any external
   notification systems.

6. Security Considerations

   Notification content (payload) needs to be protected against
   eavesdropping and alteration when it contains specific information
   from messages, such as the sender.

   Even when the content is trivial and does not contain privacy-
   sensitive information, guarding against denial-of-service attacks may
   require authentication or verification of the notification sender.

   Protocols that manipulate filters need mechanisms to protect against
   modification by, as well as disclosure to, unauthorized entities.
   For example, a malicious entity might try to delete notifications the
   user wants, or try to flood the target device with notifications to
   incur usage charges, or prevent normal use. In addition, the filters
   themselves might contain sensitive information or reveal
   interpersonal or inter-organizational relationships, as well as email
   addresses.

7. References

7.1. Normative References

   [IMAP]   Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL -
                       VERSION 4rev1", RFC 3501, March 2003.

   [LEMONADE-PROFILE]
                       Cridland, D., Ed., Melnikov, A., Ed., and S. Maes,
                       Ed., "The Internet Email to Support Diverse Service
                       Environments (Lemonade) Profile", RFC 5550, August
                       2009.

7.2. Informative References

   [BIFF]   Gellens, R., "Simple New Mail Notification", RFC 4146,
                       August 2005.

   [CONTEXT]   Cridland, D. and C. King, "Contexts for IMAP4", RFC
                       5267, July 2008.

   [CONDSTORE]   Melnikov, A. and S. Hole, "IMAP Extension for
                       Conditional STORE Operation or Quick Flag Changes
                       Resynchronization", RFC 4551, June 2006.

   [IMAP-SIEVE]   Leiba, B., "Support for Sieve in Internet Message
                       Access Protocol (IMAP4)", Work in Progress, February
                       2008.

   [MANAGE-SIEVE]   Melnikov, A., Ed., and T. Martin, "A Protocol for
                       Remotely Managing Sieve Scripts", Work in Progress,
                       September 2008.

   [MSGEVENT]   Gellens, R. and C. Newman, "Internet Message Store
                       Events", RFC 5423, March 2009.

   [IDLE]   Leiba, B., "IMAP4 IDLE command", RFC 2177, June 1997.

   [NOTIFY]   Gulbrandsen, A., King, C., and A. Melnikov, "The IMAP
                       NOTIFY Extension", RFC 5465, February 2009.

   [OMA-LEMONADE-ARCH]
                       Burger, E. and G. Parsons, "LEMONADE Architecture -
                       Supporting Open Mobile Alliance (OMA) Mobile Email
                       (MEM) Using Internet Mail", RFC 5442, March 2009.

    [OMA-ME-RD]   Open Mobile Alliance Mobile Email Requirement
                     Document, (Work in progress).
                     http://www.openmobilealliance.org/

    [SIEVE]   Guenther, P., Ed., and T. Showalter, Ed., "Sieve: An
                     Email Filtering Language", RFC 5228, January 2008.

    [SIEVE-NOTIFY] Melnikov, A., Ed., Leiba, B., Ed., Segmuller, W., and
                     T. Martin, "Sieve Email Filtering: Extension for
                     Notifications", RFC 5435, January 2009.

    [SIEVE-VARIABLES]
                     Homme, K., "Sieve Email Filtering: Variables
                     Extension", RFC 5229, January 2008.

    [URLAUTH]   Crispin, M., "Internet Message Access Protocol (IMAP)
                     - URLAUTH Extension", RFC 4467, May 2006.

8. Contributors

    The original (and longer and more detailed) version of this document
    was authored by Stephane H. Maes and Ray Cromwell of Oracle
    Corporation.

    The current and original authors want to thank all who have
    contributed key insight in notifications and filtering and have
    authored specifications or documents used in this document.

    The current and original authors want to thank the authors of the
    original work on "Server To Server Notification Protocol
    Requirements", some of whose material has been incorporated in the
    present document, in particular, Gev Decktor.

Author's Address

    Randall Gellens, Editor
    QUALCOMM Incorporated
    5775 Morehouse Drive
    San Diego, CA 92121
    USA

    EMail: rg+ietf@qualcomm.com