Abstract

This document discusses how to provide notification and filtering mechanisms to mail stores to meet Lemonade goals.

This document also discusses the use of server to server notifications, and how server to server notifications fit into an architecture which provides server to client notifications.
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 which make use of IETF work also expressed such requirements (see, for example, [OMA-LEMONADE-ARCH]; 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 [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.
In Figure 1, four categories of filters are defined:

1. **AF**: Administrative Filters: Created and maintained by mail admin. 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: Determines 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 and 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 |
|   all email    =========| email to be synched by the mobile client (CONTEXT) | ==><=== | important email / events |
+----------------+       +---------------+      |     +------------+
|            | |
| IDLE / NOTIFY | Out-of-IMAP Notifications |
| V V          | |
```

*Figure 2: Filters and Repositories*
For in-IMAP ("inband") notifications, the 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 the Section 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 all messaging systems’ accounts, such as email and voice mail), 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 indication (MWI), notification mechanisms are needed which 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 which 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, it is often 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:
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 which 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 which might deliver an equivalent message to the user or to a client.

Within the context of 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 which 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 Lemonade profile, the event order is generally not important. By including information such as the modification sequence identifier (called a modseq or mod-sequence) 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).

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 which 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 e-mail addresses.

7. IANA Considerations

None.

8. Normative References


[LEMONADE-PROFILE] D. Cridland, A. Melnikov, S. Maes, "LEMONADE
profile bis", draft-ietf-lemonade-profile-bis-08.txt, (work in
progress).

9. Informative References

2005.

[CONTEXT] D. Cridland, C. King, "Contexts for IMAP4",
draft-cridland-imap-context-05.txt (work in progress).


[IMAP-SIEVE] Leiba, B., "Support for Sieve in Internet Message
Access Protocol (IMAP4)"", draft-ietf-lemonade-imap-sieve-05.txt
(work in progress).

Managing Sieve Scripts", draft-martin-managesieve-08.txt, (work in
progress).

draft-ietf-lemonade-msgevent-05.txt, (work in progress).


[NOTIFY] C. King, A. Melnikov, A. Gulbrandsen, "The IMAP NOTIFY
Extension", draft-ietf-lemonade-imap-notify-05.txt, (work in
progress).

[OMA-LEMONADE-ARCH], E. Burger, G. Parsons, "LEMONADE Architecture
-- Supporting OMA Mobile Email (MEM) using Internet Mail",
draft-ietf-lemonade-architecture-01.txt, (work in progress).
10. 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 drafts 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 (draft-ietf-lemonade-notify-s2s-00) some of whose material has been incorporated in the present document, and in particular, Gev Decktor.

11. Authors’ Addresses

Randall Gellens
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121
rg+ietf@qualcomm.com

Stephane H. Maes
Oracle Corporation
500 Oracle Parkway
M/S 4op634
Redwood Shores, CA 94065
USA
Phone: +1-650-607-6296
Appendix A: Changes from Previous Versions

THIS SECTION TO BE REMOVED BY THE RFC EDITOR PRIOR TO PUBLICATION.
version -10:
  o Rewording of IMAP-SIEVE reference in Section 2.

version -09:
  o Reworded and moved sieve-variables reference in Section 2.
  o Reworded imap-sieve reference in Section 2.
  o Changed reference to Profile-Bis from Section 4.1.2 to "External Notifications".
  o Reworded mention of security relationship responsibility in Section 5.2.

version -08:
  o Removed mention of vacation notices in Section 2, Figure 1 (description of DF category of filters) since vacation notices, while often a function of delivery filters, are out of scope of lemonade.
  o Improved text in Section 5 and 5.1.
  o Renamed Sections 5 and 5.1.
  o Deleted old section 5.4.1 (Generic Case).
  o Reworded Section 1 (Introduction).
  o Added Section 1.1 to explain that "Lemonade Profile" here refers to profile-bis.
  o Deleted reference to original (non-bis) Profile.
  o Improved wording in Section 3.
  o Split latter part of Section 5.1 into new Section 5.2 (Scope).
  o Combined Sections 5.3 (Notification payloads) and 5.4 (Server to server notification protocol details), as the sections no longer discuss separate topics, nor were they any longer related to their titles.
  o Updated references to current versions.

version -07:
  o Fixed bugs in NF arrows in Figure 1.
  o Edited text following Figure 1 to try and make it more clear that NF can be either in the mail store (using Sieve) or in the notifications engine (using mechanisms out of scope to Lemonade).
  o Changed Acknowledgments section to Contributors to better reflect Stephan’s and Ray’s text.
  o Tweaked text in Section 5.
  o Added mention of filter protection to Security Considerations.

version -06:
  o Added pointer to profile-bis 5.4.2 ("External Notifications")
  o Cleaned up references, split into Normative/Informative
"inband" and "outband".

- Added mention of user login/logout as a state change that can be used to trigger outband notifications.
- Redid Figure 1 to make it easier to understand, and also to show that NF might be within the mail store (Sieve) or in an external notification mechanism.
- Deleted section 5.3 ("Server to server terminology").
- Deleted reference to draft-ietf-lemonade-notification-protocol-xx.
- Major changes to section 5.4 ("Notification payloads") and section 5.5 ("Server to server notification protocol details").
- Deleted section 5.5.2 ("Abstracted notification protocol"), section 5.5.3 ("Exception Handling"), and section 5.6 ("Server to server complementary information").
- Rewrote section 5.7 ("Event orders") and added reference to RFC 4551.
- Rewrote section 5.8 ("Reliability").
- Added references to sieve-notify, IMAP NOTIFY, IMAP, and others.
- Deleted many, many references.
- Updated references.
- Corrected references.
- Split references into normative and informative.
- Added reference to draft-ietf-lemonade-architecture-00.txt.
- Rewrote Introduction.
- Changed draft name from "... and Filters" to "... Architecture".

version -05:
- Significant deletion of sections, per www1.ietf.org/mail-archive/web/lemonade/current/msg03936.html

version -04:
- Update dates, slight reformatting, add editor’s note for References

version -03:
- Updated examples to use new METADATA syntax
- Drop CLEARIDLE and reference A. Melnikov’s IMAP-EVENTS
- XEMN notification format extended to with event and view attributes

- View filter is a work in progress. Several proposals are being discussed, so the draft has been revised to try and capture high level requirements (e.g. out of band notifications must be able to identify which view an event occurred for)
- Added notification protocol details and reference

version -02:
- LPROVISION/LGETPREFS/LSETPREFS removed in favor of mailbox annotations
- Updated inband notification section to include discussion of CLEARIDLE and MSGEVENTS
- EMN payload clarified for both wakeup and extended formats.
- Some reference clean-up
- Add server to server notifications based on the expired draft
version -01:
- Move SMS / WAP examples to an informative appendix.
- Restrict the exchange of keys via LPROVISION to secure exchanges.
- Differentiate ANNOTATE from LPROVISION on that basis.

version -00:
- Initial release

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement
Funding for the RFC Editor function is currently provided by the Internet Society.