Groupe de travail Réseau
**Request for Comments : 5550**
RFC rendue obsolète : 4550
RFC mises à jour : 4469, 4467
Catégorie : Sur la voie de la normalisation

D. Cridland, éd., Isode Limited
A. Melnikov, éd., Isode Limited
S. Maes, éd., Oracle
août 2009
Traduction Claude Brière de L'Isle

# Profil de messagerie Internet pour la prise en charge de divers environnements de service (Lemonade)

**Résumé**
Le présent document décrit un profil (ensemble d'extensions, restrictions, et modes d'usage exigés) appelé Lemonade, des protocoles IMAP, de soumission de messagerie, et Sieve. Ce profil permet aux clients (en particulier ceux qui ont des contraintes de mémoire, de bande passante, de puissance de calcul, ou autres) d'utiliser efficacement IMAP et le protocole de soumission pour accéder et soumettre des messages. Cela inclut la capacité de transmettre les messages reçus sans avoir besoin de les télécharger, pour optimiser la soumission, et pour se resynchroniser efficacement en cas de perte de connexité avec le serveur.

Le profil Lemonade s'appuie sur plusieurs extensions aux protocoles IMAP, Sieve, et de soumission de messagerie. Le document définit aussi une nouvelle extension IMAP et enregistre plusieurs nouveaux mots-clés IMAP.

**Statut de ce mémoire**
Ceci est un document de l'Internet sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc5550.



## Table des matières

1. Introduction....2
2. Conventions utilisées dans ce document....3
3. Résumé de la prise en charge exigée....3
3.1 Serveurs de soumission Lemonade....3
3.2 Mémorisations de message Lemonade....3
3.3 Agents de livraison de message Lemonade....4
4. Serveurs de soumission Lemonade....4

4.1 Transmission sans téléchargement....5
4.2 Traitement en parallèle....5
4.3 Prise en charge du DSN....5
4.4 Déclaration de taille de message....5
4.5 Prise en charge de codes d'état améliorés....5
4.6 Chiffrement et compression....5
5. Mémorisations de messages Lemonade....6
5.1 Resynchronisation rapide....6
5.2 Traitement de la partie message....6
5.3 Compression....6
5.4 Notifications....6
5.5 Filtres de recherche et de vue....7
5.6 Traitement de boîte aux lettres....7
5.7 Transmission sans téléchargement....8
5.8 Extensions IMAP supplémentaires....8
5.9 Enregistrement du mot-clé IMAP $Forwarded....8
5.10 Enregistrement des mots-clés IMAP $SubmitPending et $Submitted....8
5.11 Extensions IMAP en rapport....9
6. Agents de livraison de message Lemonade....9
7. Agents d'utilisateur de message Lemonade....9
8. Transmission sans téléchargement....10
8.1 Motivations....10
8.2 Vue d'ensemble de l'envoi de message....10
8.3 Stratégie traditionnelle....10
8.4 Nouvelle stratégie....10
8.5 Considérations sur la sécurité des Pawn-Tickets....17
8.6 Copies des messages envoyés : le problème de fcc....17
9. Considérations de déploiement....17
10. Considérations sur la sécurité....17
10.1 Protection de la confidentialité des messages soumis....17
10.2 TLS....18
10.3 Extensions supplémentaires et modèles de déploiement....18
11. Considérations relatives à l'IANA....18
12. Changements par rapport à la RFC 4550....18
13. Remerciements....19
14. Références....19
14.1 Références normatives....19
14.2 Références pour information....21
Appendice A. Errata....22
Adresse des éditeurs....25

## 1. Introduction

Le profil Lemonade, ou simplement Lemonade, fournit des améliorations à la messagerie Internet pour prendre en charge divers environnements de service. Les serveurs de messagerie Lemonade fournissent à la fois un serveur de soumission Lemonade et une mémorisation de message Lemonade, qui se fondent respectivement sur les protocoles existants des [RFC4409] et [RFC3501]. Ils PEUVENT aussi inclure un agent de livraison de message Lemonade, qui fournit des services de filtrage de l'heure de livraison fondés sur la [RFC5228].

Le présent document décrit le profil Lemonade qui inclut :

- Des améliorations communes générales à la messagerie Internet Mail, décrites aux Sections 4 et 5.
- "Transmission sans téléchargement" qui décrit les échanges entre clients et serveurs Lemonade pour permettre de soumettre de nouveaux messages électroniques incorporant du contenu qui réside dans des localisations externes au client, à la Section 8.
- La resynchronisation rapide de boîte aux lettres, décrite au paragraphe 5.1.
- Les extensions pour prendre en charge des notifications plus précises, et plus larges, provenant de la mémorisation dans la prise en charge des notifications et des filtres de vue, décrites aux paragraphes 5.4.1 et 5.5.
- Le filtrage de l'heure de livraison dans la prise en charge des cas d'utilisation normaux de gestion de messagerie, comme décrit au paragraphe 3.3.

Le groupe de travail LEMONADE a utilisé l'architecture montrée dans la [RFC5442] pour développer le profil Lemonade.

Il est prévu que le profil Lemonade prenne en charge les réalisations de l'utilitaire de messagerie mobile (MEM, *mobile email enabler*) de l'OMA (voir [OMA-MEM-REQ] et [OMA-MEM-ARCH]) en utilisant les protocoles de messagerie Internet définis par l'IETF.

## 2. Conventions utilisées dans ce document

Dans les exemples, "M:", "I:", et "S:" indiquent les lignes envoyés respectivement par l'agent d'utilisateur de message de client, le serveur de messagerie IMAP, et le serveur de soumission SMTP.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Les autres mots en majuscules sont normalement les noms des extensions ou des commandes -- celles-ci ne sont en majuscules que pour la lisibilité, et sont insensibles à la casse.

Le présent document utilise la terminologie définie dans la [RFC5598]. Voir dans la [RFC5598] les détails de l'architecture de la messagerie électronique.

Tous les exemples de ce document sont optimisés pour l'usage de Lemonade et pourraient ne pas représenter des exemples d'usage approprié du protocole pour un client d'utilisation générale Submit/IMAP. En particulier, les exemples supposent que les serveurs Submit et IMAP prennent en charge toutes les extensions Lemonade décrites dans ce document, donc ils ne montrent pas de replis en l'absence d'une extension.

## 3. Résumé de la prise en charge exigée

### 3.1 Serveurs de soumission Lemonade

Les serveurs de soumission Lemonade DOIVENT fournir un service comme décrit dans la [RFC4409], et DOIVENT prendre en charge ce qui suit. Noter que le profil Lemonade impose plus d'exigences pour certains cas, précisés dans les paragraphes cités.

| Extension SMTP | Référence | Exigences |
|---|---|---|
| 8BITMIME | [RFC1652] | [RFC4468] |
| AUTH | [RFC4954] | [RFC4409] |
| BINARYMIME | [RFC3030] | paragraphe 4.1 |
| BURL imap | [RFC4468] | Section 8 |
| CHUNKING | [RFC3030] | paragraphe 4.1 |
| DSN | [RFC3461] | paragraphe 4.3 |
| ENHANCEDSTATUSCODES | [RFC2034] | paragraphe 4.5 |
| PIPELINING | [RFC2920] | paragraphe 4.2 |
| SIZE | [RFC1870] | paragraphe 4.4 |
| STARTTLS | [RFC3207] | paragraphe 4.6 |

### 3.2 Mémorisations de message Lemonade

Les mémorisations de message DOIVENT fournir un service comme décrit dans la [RFC3501], et DOIVENT prendre en charge ce qui suit. Noter que le profil Lemonade impose plus d'exigences dans certains cas, détaillées dans les paragraphes cités.

| Extension IMAP | Référence | Exigences |
|---|---|---|
| BINARY | [RFC3516] | paragraphe 5.2 |
| CATENATE | [RFC4469] | paragraphe 5.7 |
| COMPRESS=DEFLATE | [RFC4978] | paragraphe 5.3 |

| | | |
|---|---|---|
| CONDSTORE | [RFC4551] | paragraphe 5.1 |
| CONTEXT=SEARCH | [RFC5267] | paragraphe 5.5 |
| CONTEXT=SORT | [RFC5267] | paragraphe 5.5 |
| CONVERT | [RFC5259] | paragraphe 5.2 |
| ENABLE | [RFC5161] | paragraphe 5.1 |
| ESEARCH | [RFC4731] | paragraphe 5.5 |
| ESORT | [RFC5267] | paragraphe 5.5 |
| I18NLEVEL=1 | [RFC5255] | paragraphe 5.8 |
| IDLE | [RFC2177] | paragraphe 5.4.1 |
| LITERAL+ | [RFC2088] | paragraphe 5.8 |
| NAMESPACE | [RFC2342] | paragraphe 5.6 |
| NOTIFY | [RFC5465] | paragraphe 5.4.1 |
| QRESYNC | [RFC5162] | paragraphe 5.1 |
| SASL-IR | [RFC4959] | paragraphe 5.8 |
| SORT | [RFC5256] | paragraphe 5.5 |
| STARTTLS | [RFC3501] | - |
| UIDPLUS | [RFC4315] | paragraphe 5.7 |
| URLAUTH | [RFC4467] | paragraphe 5.7 |
| URL-PARTIAL | Section 5.7.1 | paragraphe 5.7 |
| Mot clé $Forwarded | - | paragraphe 5.9 |
| Mot clé $SubmitPending | - | paragraphe 5.10 |
| Mot clé $Submitted | - | paragraphe 5.10 |

En plus de cette liste, toute mémorisation de message Lemonade DOIT envoyer le code de réponse CAPABILITY (voir le paragraphe 7.1 de la [RFC3501]) dans l'accueil initial du serveur et après les commandes LOGIN/AUTHENTICATE.

### 3.3 Agents de livraison de message Lemonade

Les agents de livraison de message Lemonade DOIVENT prendre en charge le langage de filtrage de messagerie Sieve comme décrit dans la [RFC5228], et DOIVENT prendre en charge les extensions Sieve suivantes. Noter que le profil Lemonade impose plus d'exigences dans certains cas, détaillés dans les paragraphes cités.

| **Extension Sieve** | **Référence** | **Exigences** |
|---|---|---|
| ENOTIFY | [RFC5435] | Section 6 |
| IMAP4FLAGS | [RFC5232] | Section 6 |
| RELATIONAL | [RFC5231] | Section 6 |
| VACATION | [RFC5230] | Section 6 |
| VARIABLES | [RFC5229] | Section 6 |
| comparator-i;unicode-casemap | [RFC5051] | Section 6 |

Les agents de livraison de message Lemonade devraient aussi considérer de prendre en charge un protocole de gestion de script Sieve, comme celui de la [RFC5804].

## 4. Serveurs de soumission Lemonade

Tous les serveurs de soumission Lemonade mettent en œuvre le protocole de soumission de messagerie décrit dans la [RFC4409], qui est à son tour un profil spécifique de la [RFC5321]. Donc, tout MUA conçu pour soumettre de la messagerie via la [RFC4409] ou la [RFC5321] va interopérer avec les serveurs de soumission Lemonade.

De plus, les serveurs de soumission Lemonade mettent en œuvre l'ensemble suivant de SMTP et d'extensions de soumission pour accroître l'efficacité de la soumission de message.

### 4.1 Transmission sans téléchargement

Afin d'optimiser l'usage du réseau pour le cas normal où le contenu de message est copié de, ou a pour source, la mémorisation IMAP, Lemonade fournit la prise en charge d'une suite d'extensions appelées collectivement la "transmission sans téléchargement", discutée en détails à la Section 8.

Les serveurs de soumission Lemonade DOIVENT prendre en charge les extensions SMTP BURL [RFC4468], 8BITMIME [RFC1652], BINARYMIME [RFC3030], et CHUNKING [RFC3030].

BURL DOIT prendre en charge les URL de type URLAUTH [RFC4467], et donc DOIT annoncer l'option "imap" suivant le mot-clé BURL EHLO (voir les détails dans la [RFC4468]).

### 4.2 Traitement en parallèle

Certains clients utilisent régulièrement les réseaux avec une latence relativement élevée, comme les réseaux mobiles ou fondés sur le satellite. Éviter des allers-retours au sein d'une transaction a un grand avantage pour la réduction de bande passante comme pour le temps total de transaction. Pour cette raison, les serveurs de soumission de messagerie conformes à Lemonade DOIVENT prendre en charge les extensions de service SMTP pour l'exécution en parallèle de commandes [RFC2920].

### 4.3 Prise en charge de DSN

Les serveurs de soumission de messagerie conformes à Lemonade DOIVENT prendre en charge les extensions de service SMTP pour les notifications d'état de livraison [RFC3461].

### 4.4 Déclaration de taille de message

Il y a souvent un avantage particulier à détecter les cas de défaillance aussitôt que possible, comme lorsque l'utilisateur est facturé par octet, ou quand la bande passante est faible. C'est particulièrement vrai des messages de grande taille.

Les serveurs de soumission Lemonade DOIVENT prendre en charge l'extension de service SMTP pour la déclaration de taille de message [RFC1870].

Les serveurs de soumission Lemonade DOIVENT développer toutes les parties BURL avant d'évaluer si la taille de message fournie est acceptable.

Un client à capacité Lemonade DEVRAIT utiliser la déclaration de taille de message. En particulier, le client NE DOIT PAS envoyer un message à un serveur de soumission de messagerie si il sait que le message excède la taille maximale de message annoncée par le serveur de soumission. Quand il inclut une taille de message dans la commande MAIL FROM, le client DOIT utiliser une valeur au moins aussi grande que la taille des données de message assemblées après la résolution de toutes les parties BURL.

### 4.5 Prise en charge de codes d’état améliorés

Les serveurs de soumission de messagerie conformes à Lemonade DOIVENT prendre en charge l'extension de service SMTP pour retourner des codes d'erreur améliorés [RFC2034]. Cela permet à un client de déterminer la cause précise de défaillance.

### 4.6 Chiffrement et compression

Les serveurs de soumission de messagerie conformes à Lemonade DOIVENT prendre en charge l'extension de service SMTP pour SMTP sûr sur la sécurité de la couche transport (TLS, *Transport Layer Security*) [RFC3207].

La prise en charge de la méthode de compression DEFLATE, décrite dans la [RFC3749], est RECOMMANDÉE.

## 5. Mémorisations de messages Lemonade

Toutes les mémorisations de message Lemonade mettent en œuvre le protocole d'accès au message Internet (IMAP, *Internet Message Access Protocol*) défini dans la [RFC3501]. Donc, tout MUA écrit pour accéder aux messages en utilisant les facilités décrites dans la [RFC3501] va interopérer avec une mémorisation de message Lemonade.

De plus, les mémorisations de message Lemonade fournissent un ensemble d'extensions pour traiter les limitations de

certains clients et réseaux.

### 5.1 Resynchronisation rapide

La resynchronisation est une partie coûteuse d'une session IMAP, et les réseaux mobiles sont généralement plus enclins à des déconnexions involontaires, ce qui à son tour rend ce problème plus aigu. Donc, les mémorisations de message Lemonade fournissent une suite d'extensions pour réduire le coût de synchronisation.

Les serveurs IMAP conformes à Lemonade DOIVENT prendre en charge les extensions CONDSTORE [RFC4551], QRESYNC [RFC5162], et ENABLE [RFC5161]. Cela permet à un client de resynchroniser rapidement toute boîte aux lettres en demandant au serveur de retourner tous les changements de fanions et de purger ceux qui se sont produits depuis un état précédemment enregistré. Cela peut aussi accélérer la reconnexion d'un client dans le cas où la couche de transport est coupée, accidentellement ou au titre d'un changement dans le réseau.

Quand ils mettent en œuvre QRESYNC [RFC5162], les clients et serveurs doivent aussi se conformer aux errata soumis pour ce document (voir l'Appendice A).

La [RFC4549] précise comment les clients effectuent une resynchronisation efficace de boîte aux lettres.

### 5.2 Traitement de la partie message

Le traitement des parties de message, particulièrement des pièces jointes, représente un ensemble de défis pour les appareils limités, en termes de bande passante utilisée et de capacités de l'appareil.

Les serveurs IMAP conformes à Lemonade DOIVENT prendre en charge l'extension BINARY [RFC3516]. Cela déplace les opérations de décodage des parties de corps MIME du client au serveur. Les données décodées sont de taille égale ou inférieure à la représentation codée, donc cela réduit effectivement la bande pasante.

La [RFC3516] permet aux serveurs de refuser d'accepter les messages téléchargés qui contiennent des données binaires, en n'acceptant pas le codage de transfert de contenu "Binary" ; cependant, les serveurs IMAP conformes à Lemonade DEVRONT toujours accepter les messages MIME codés en binaire dans les commandes APPEND pour tout fichier.

La [RFC5259] DOIT aussi être prise en charge par les serveurs, ce qui permet aux clients de demander des conversions entre types de supports, et permet d'adapter les images, etc. Cela donne la capacité de voir les pièces jointes (et parfois des parties de corps) sans la facilité de traiter une large gamme de types de supports, ou de voir efficacement les pièces jointes.

### 5.3 Compression

Les mémorisations de message Lemonade DEVRAIENT prendre en charge l'algorithme de compression Deflate pour TLS, comme défini dans la [RFC3749], afin de faciliter la compression au plus bas niveau possible.

Cependant, le groupe de travail reconnaît que pour de nombreux points d'extrémité, c'est une technologie rarement déployée, et à ce titre, les mémorisations de message Lemonade DOIVENT fournir [RFC4978] la prise en charge d'une compression de flux de repli au niveau de l'application, lorsque TLS ne fournit pas activement la compression.

### 5.4 Notifications

L'ajout de notifications de serveur à client transforme le profil Lemonade en un protocole de synchronisation fondé sur l'événement. Chaque fois que se produit un événement qui intéresse le MUA, une notification peut être générée. Le groupe de travail Lemonade a utilisé l'architecture de notifications montrée dans la [RFC5551] pour développer le profil Lemonade.

Si le MUA est connecté au serveur IMAP, les notifications dans la bande sont générées en utilisant les facilités mentionnées au paragraphe 5.4.1.

Quand le MUA n'est pas connecté, le filtre de notification génère une notification hors bande. Le filtre de notification peut être considéré comme agissant comme un répertoire de poussée de messagerie.

Si le MUA n'est pas connecté, et si la notification hors bande est désactivée, le client doit effectuer une synchronisation rapide à la reconnexion pour déterminer les changements de boîte aux lettres, en utilisant les mécanismes du paragraphe 5.1.

### 5.4.1 Notifications IMAP

Les mémorisations de message Lemonade DOIVENT prendre en charge l'extension IDLE [RFC2177]. L'extension permet aux clients de recevoir des notifications non sollicitées sur les changements de la boîte aux lettres choisie, sans qu'il soit besoin d'interroger sur les changements. Les réponses qui forment ces notifications DOIVENT être envoyées en temps utile quand de tels changements se produisent.

Les mémorisations de message Lemonade fournissent aussi l'extension NOTIFY décrite dans la [RFC5465], qui permet aux clients de demander des types spécifiques d'événement à envoyer immédiatement au client, pour le dossier actuellement choisi et les autres. Ces types d'événement incluent la livraison de message et les changements de nom de boîte aux lettres.

### 5.4.2 Notifications externes

Lemonade et TCP assurent des connexions au repos de longue durée entre le client et la mémorisation de messagerie, permettant au serveur de pousser les notifications dans IMAP. Certains réseaux mobiles prennent en charge la dormance, qui ferme le canal de trafic radio durant les périodes de repos pour conserver les ressources du combiné et du réseau, tout en conservant l'état IP et TCP. (Voir la [RFC5383] pour plus d'informations.)

Cependant, il y a des environnements où le client de messagerie ne peut pas rester actif indéfiniment, ou où il n'est pas conseillé (ou même pas toujours possible) que les connexions TCP au serveur restent en service sans activité pendant de longues périodes. Dans ces situations, une bonne expérience d'utilisateur exige que quand des événements"intéressants" se produisent dans la mémorisation de messagerie, le client soit informé pour qu'il puisse se connecter et se resynchroniser. Au minimum, cela exige qu'au moins l'arrivée d'un nouveau message génère une sorte d'éveil du client de messagerie. Un certain nombre de fabricants ont mis en œuvre diverses solutions pour cela. Comme exemples de ce qui a été fait, pendant des années (longtemps avant les téléphones cellulaires) la technique décrite dans la [RFC4146] a été prise en charge. Aujourd'hui, un certain nombre d'entreprises de messagerie électronique incluent des facilités pour envoyer des SMS ou autre simples messages non en flux aux clients sur les appareils quand un nouveau message arrive. L'alliance mobile ouverte (OMA; *Open Mobile Alliance*) a publié un mécanisme qui utilise WAP PUSH pour envoyer un message basique contenant un URL [OMA-EMN]. L'IETF cherche les moyens de normaliser une fonctionnalité amélioré dans ce domaine.

Une expérience d'utilisateur de "messagerie poussée" peut être réalisée en utilisant diverses techniques, allant de la connexité TCP toujours activée au serveur et l'extension NOTIFY décrite ci-dessus, à OMA EMN, ou même un message non standard de déclenchement sur un SMS. Dans toute technique, le client apprend l'existence d'un nouveau message, et décide d'aller chercher des informations sur lui, une partie de lui, ou sa totalité, et ensuite de le présenter à l'utilisateur.

## 5.5 Filtres de recherche et de vue

Les mémorisations de message Lemonade DOIVENT prendre en charge l'extension ESEARCH [RFC4731]. L'extension permet aux clients de trouver effectivement le premier ou le dernier message, de trouver un compte de messages correspondants, et d'obtenir une liste de messages correspondants dans une représentation considérablement plus compacte.

Les mémorisations de message Lemonade fournissent aussi un mécanisme pour que les clients évitent de traiter une boîte aux lettres entière, au lieu d'accéder à une vue de la boîte aux lettres. Cette technique, courante dans de nombreux clients d'ordinateurs comme une capacité côté client, est utile pour les clients contraints pour minimiser la quantité de messages et de données de notification.

Les mémorisations de message Lemonade DOIVENT donc mettre en œuvre les extensions CONTEXT=SEARCH, ESORT, et CONTEXT=SORT définies dans la [RFC5267], ainsi que l'extension SORT définie dans la [RFC5256].

## 5.6 Traitement de boîte aux lettres

Les mémorisations de message Lemonade DOIVENT prendre en charge l'extension NAMESPACE [RFC2342]. L'extension permet aux clients de découvrir des boîtes aux lettres partagées et des boîtes aux lettres appartenant à d'autres

utilisateurs, et fournit une vue hiérarchique normalisée des boîtes aux lettres disponibles.

Les mémorisations de message Lemonade DOIVENT prendre en charge l'extension I18NLEVEL=<n> [RFC5255], avec <n> ayant la valeur 1 ou 2. Cela ajoute la prise en charge de fonctions de recherche et de tri non en anglais (internationalisées). (Noter que I18NLEVEL=2 implique de prendre en charge I18NLEVEL=1, de sorte qu'un client conforme à Lemonade qui utilise cette extension DOIT les reconnaître toutes les deux.)

### 5.7 Transmission sans téléchargement

Afin d'optimiser l'usage du réseau pour le cas normal où le contenu de message est copié de, ou a pour source, la mémorisation IMAP, Lemonade fournit la prise en charge d'une suite d'extensions collectivement appelés la "transmission sans téléchargement", discutée en détails à la Section 8.

Les mémorisations de message Lemonade DOIVENT prendre en charge CATENATE [RFC4469], UIDPLUS [RFC4315], et URLAUTH [RFC4467]. Les mémorisations de message Lemonade DOIVENT aussi prendre en charge URL-PARTIAL comme décrit au paragraphe 5.7.1.

#### 5.7.1 Prise en charge de PARTIAL dans CATENATE et URLAUTH

La [RFC5092] a introduit un nouvel élément syntaxique pour faire référence à une gamme d'octets d'un message/partie de corps. C'est fait en utilisant le champ ;PARTIAL=. Si un serveur IMAP prend en charge PARTIAL dans l'URL IMAP utilisé dans les extensions CATENATE et URLAUTH, il DOIT alors annoncer la capacité URL-PARTIAL dans la réponse CAPABILITY et dans le code de réponse équivalent.

### 5.8 Extensions IMAP supplémentaires

Les mémorisations de message Lemonade DOIVENT prendre en charge l'extension LITERAL+ [RFC2088]. L'extension permet aux clients d'économiser un aller-retour chaque fois qu'un littéral non synchronisant est envoyé.

Les mémorisations de message Lemonade DOIVENT aussi mettre en œuvre l'extension SASL-IR [RFC4959], qui permet aux clients d'économiser un aller-retour durant l'authentification, traitant potentiellement en parallèle la séquence d'authentification entière.

Les serveurs IMAP conformes à Lemonade DOIVENT prendre en charge IMAP sur TLS [RFC3501] comme exigé par la [RFC3501]. Comme noté au paragraphe 5.3, les serveurs DEVRAIENT prendre en charge l'algorithme de compression deflate pour TLS, comme spécifié dans la [RFC3749].

### 5.9 Enregistrement du mot-clé IMAP $Forwarded

Le mot-clé IMAP $Forwarded est utilisé par plusieurs clients IMAP pour spécifier que le message marqué a été transmis à une autre adresse de messagerie, incorporée dans ou jointe à un nouveau message. Un client de messagerie établit ce mot-clé quand il réussit à transmettre le message à une autre adresse de messagerie. L'usage normal de ce mot-clé est de montrer une icône différente (ou supplémentaire) pour un message qui a été transmis. Une fois établi, le fanion NE DEVRAIT PAS être mis à zéro.

Les mémorisations de message Lemonade DOIVENT être capables de mémoriser le mot-clé $Forwarded. Elles DOIVENT le préserver sur l'opération COPY. Les serveurs DOIVENT prendre en charge le SEARCH KEYWORD $Forwarded.

### 5.10 Enregistrement des mots-clés IMAP $SubmitPending et $Submitted

Le mot-clé IMAP $SubmitPending désigne le message comme attendant d'être soumis. Ce mot-clé permet de mémoriser des messages qui attendent d'être soumis dans la même boîte aux lettres où les messages qui ont déjà été soumis et/ou sont en cours d'édition sont mémorisés. Un client de messagerie établit ce mot-clé quand il décide que le message doit être envoyé. Quand un client (il pourrait être un client différent de celui qui a décidé que le message est en attente de soumission) commence à envoyer le message, il ajoute de façon atomique (en utilisant "STORE (UNCHANGEDSINCE)") le mot-clé $Submitted. Une fois la soumission réussie, le mot-clé $SubmitPending est supprimé. Les deux mots-clés permettent que les messages qui sont soumis de façon active (les messages qui ont les deux mots clés $Submitted et $SubmitPending établis) soient distingués des messages qui attendent d'être soumis, ou des messages déjà soumis. Cela

permet aussi que tous les messages qui étaient supposés devoir être soumis soient trouvés, si le client qui les soumet a une panne ou quitte avant de les soumettre.

Les mémorisations de message Lemonade DOIVENT être capables de mémoriser les mots-clés $SubmitPending et $Submitted. Les mémorisations de message Lemonade DOIVENT les préserver sur l'opération COPY. Les serveurs DOIVENT prendre en charge SEARCH KEYWORD $SubmitPending et SEARCH KEYWORD $Submitted.

### 5.11 Extensions IMAP en rapport

Le paragraphe 5.11 n'est pas normatif.

Les mises en œuvre de serveur qui visent à satisfaire les exigences de OMA MEM [OMA-MEM-REQ] devraient envisager de mettre en œuvre la [RFC5466], qui fournit un moyen d'appliquer une définition de boîtes aux lettres virtuelle sur le serveur. Elles devraient aussi envisager de mettre en œuvre l'extension METADATA-SERVER [RFC5464], qui donne un moyen de mémoriser des données définies par l'utilisateur associées à un compte d'utilisateur.

## 6. Agents de livraison de message Lemonade

Les agents de livraison de message Lemonade DOIVENT prendre en charge le langage de filtrage de la [RFC5228] au point de livraison, permettant à l'utilisateur de contrôler quels messages sont acceptés, et où ils sont mémorisés.

Les agents de livraison de message Lemonade DOIVENT prendre en charge l'extension Sieve Vacation [RFC5230], qui permet au client d'établir un auto-répondeur, normalement pour dire qu'il est en vacances (d'où le nom de l'extension Sieve).

Les agents de livraison de message Lemonade DOIVENT prendre en charge l'extension Sieve Enotify [RFC5435], qui permet à un script Sieve de générer des notifications (comme XMPP, SIP, ou de messagerie) sur les messages reçus.

Les agents de livraison de message Lemonade DOIVENT prendre en charge l'extension Sieve Variables [RFC5229], qui ajoute la prise en charge de variables du langage de scripts Sieve. Cette extension est normalement utilisée avec Sieve Enotify ou Vacation pour personnaliser les réponses/notifications.

Les agents de livraison de message Lemonade DOIVENT prendre en charge l'extension Sieve Relational [RFC5231], qui ajoute la prise en charge de comparaisons relationnelles au langage de scripts Sieve. Cette extension est normalement utilisée avec Sieve Enotify.

Les agents de livraison de message Lemonade DOIVENT prendre en charge l'extension Sieve Imap4Flags [RFC5232], qui permet à un script Sieve d'établir des fanions/mots-clés IMAP lors de la livraison d'un message à une boîte aux lettres. Par exemple, cela peut être utilisé pour marquer automatiquement certains messages comme intéressants, urgents, etc.

Les agents de livraison de message Lemonade DOIVENT prendre en charge le comparateur i;unicode-casemap dans Sieve [RFC5051], qui est déclaré comme "comparator-i;unicode-casemap" dans la déclaration Sieve "require". Le comparateur permet une comparaison insensible à la casse des caractères Unicode.

Les agents de livraison de message Lemonade devraient envisager de prendre en charge la gestion de script Sieve en utilisant le protocole de la [RFC5804]. Si ils le font, ils DOIVENT aussi annoncer toutes les extensions Sieve mentionnées dans cette section.

## 7. Agents d’utilisateur de message Lemonade

Bien que tous les MUA IMAP existants soient conformes à Lemonade dans la mesure où tous les services Lemonade sont fondés sur les protocoles existants des [RFC3501] et [RFC4409], les mises en œuvre de client sont encouragées à tirer complètement parti des facilités fournies par les serveurs de soumission et mémorisations de message Lemonade, comme décrit respectivement aux Sections 4 et 5.

Quand ils ouvrent une connexion au serveur de soumission, les clients DOIVENT le faire en utilisant l'accès 587 sauf si ils sont explicitement configurés à utiliser un autre accès [RFC5068]. (Noter que cette exigence est un peu plus forte que celle

spécifiée dans la [RFC4409], car la [RFC4409] ne prescrit pas la procédure exacte à utiliser par les client de soumission.) Si la connexion TCP au serveur de soumission échoue à s'ouvrir en utilisant l'accès 587, le client PEUT alors réessayer immédiatement en utilisant un accès différent, comme le 25. Voir la [RFC4409] pour des informations sur la raison pour laquelle l'utilisation de l'accès 25 va probablement échouer selon la localisation courante du client, et peut résulter en un code d'échec durant la transaction SMTP.

De plus, certaines spécifications sont utiles pour prendre en charge une messagerie interopérable avec une expérience d'utilisateur améliorée.

Les clients à capacité Lemonade DEVRAIENT prendre en charge les paramètres Format et DelSp au type de support text/plain décrit dans la [RFC3676], et générer ce format pour les messages.

Les clients à capacité Lemonade DEVRAIENT prendre en charge, et utiliser, le mot-clé $Forwarded décrit au paragraphe 5.9.

## 8. Transmission sans téléchargement

### 8.1 Motivations

L'arrivée de la messagerie de client/serveur en utilisant les protocoles des [RFC3501] et [RFC4409] a changé ce qui était anciennement des opérations locales de disque pour devenir des transmissions de données répétitive du réseau.

La "transmission sans téléchargement" Lemonade utilise l'extension de la [RFC4468] pour permette l'accès à des sources externes durant la soumission d'un message. En combinaison avec l'extension de la [RFC4467], l'inclusion de parties de message ou même de messages entiers provenant de la mémorisation de messagerie IMAP est possible avec une relation de confiance minimale entre les serveurs IMAP et SMTP SUBMIT.

La "transmission sans téléchargement" Lemonade a l'avantage de maintenir un protocole de soumission, et donc évite le risque d'avoir plusieurs mécanismes parallèles et possiblement divergents pour la soumission. Le client peut utiliser les extensions de la [RFC4409] sans qu'elles soient ajoutées à IMAP. De plus, en gardant les détails de la soumission de message dans le serveur SMTP SUBMIT, la "transmission sans téléchargement" Lemonade peut fonctionner avec d'autres protocoles de restitution de message comme POP, NNTP, ou ce qui pourrait être conçu à l'avenir.

### 8.2 Vue d’ensemble de l’envoi de message

L'acte d'envoyer un message électronique peut être vu comme impliquant plusieurs étapes : initiation d'un nouveau projet, édition du projet, assemblage du message, et soumission du message.

L'initiation d'un nouveau projet et l'édition du projet a lieu dans le MUA. Fréquemment, les utilisateurs choisissent de sauvegarder les messages les plus complexes sur un serveur de la [RFC3501] (via la commande APPEND avec le fanion \Draft) pour un rappel ultérieur par le MUA et la reprise du processus d'édition.

L'assemblage de message est le processus de production d'un message complet à partir de la révision finale du projet et des sources externes. Au moment de l'assemblage, les données externes sont restituées et insérées dans le message.

La soumission du message est le processus d'insertion du message assemblé dans l'infrastructure de la [RFC5321], normalement en utilisant le protocole de la [RFC4409].

### 8.3 Stratégie traditionnelle

Traditionnellement, les messages sont initiés, édités, et assemblés entièrement au sein d'un MUA, bien que les projets puissent être sauvegardés sur un serveur de la [RFC3501] et restitués plus tard du serveur. Le texte complété est alors transmis pour livraison à un agent de soumission de message (MSA, *Message Submission Agent*).

Il n'y a souvent pas de limite claire entre les processus d'édition et d'assemblage. Si un message est transmis, son contenu est souvent restitué immédiatement et inséré dans le texte du message. De même, quand un contenu externe est inséré ou joint, le contenu est généralement restitué immédiatement et intégré au projet.

Par conséquent, chaque sauvegarde d'un projet et les restitutions ultérieures du projet transmettent tout ce contenu

(éventuellement grand) comme le fait la soumission de message.

Dans le passé, cela ne posait pas de problème, parce que les projets, les données externes, et le mécanisme de soumission de message étaient normalement situés sur le même système que le MUA. Le problème le plus courant était celui du dépassement du quota du disque.

### 8.4 Nouvelle stratégie

Le modèle distingue entre un agent d'utilisateur de message (MUA, *Message User Agent*), un serveur IMAPv4Rev1 [RFC3501], et un serveur de soumission SMTP [RFC4409], comme illustré à la Figure 1.

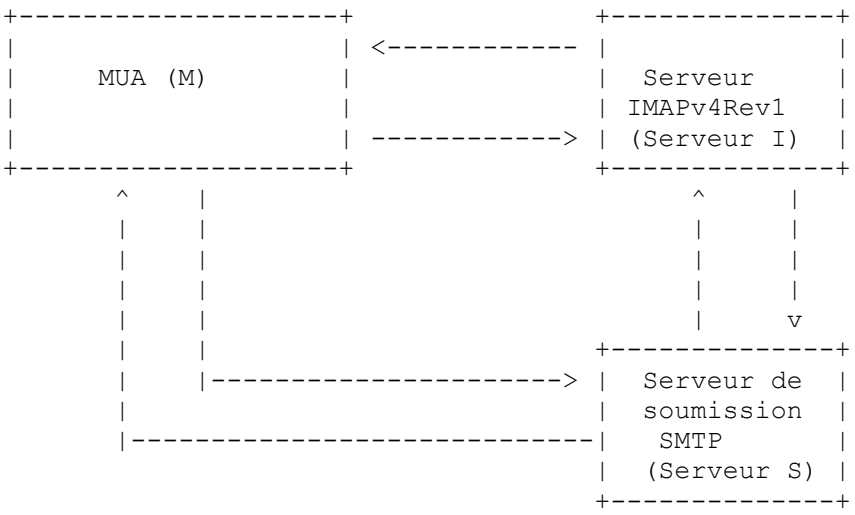


**Figure 1 : "transmission sans téléchargement" Lemonade**

La "transmission sans téléchargement" Lemonade permet à un agent d'utilisateur de message de composer et transmettre un message électronique combinant des fragments qui sont situés dans un serveur IMAP, sans avoir à télécharger ces fragments au client.

Ce paragraphe pour information décrit deux façons d'effectuer la "transmission sans téléchargement" selon l'endroit où l'assemblage du message a lieuu. La première utilise la commande APPEND étendue [RFC4469] pour éditer un projet de message dans la mémorisation de message et causer l'assemblage du message au serveur IMAP. Ceci est le plus souvent utilisé quand une copie du message est à conserver sur le serveur IMAP, comme discuté au paragraphe 8.6.

La seconde utilise une succession de commandes BURL et BDAT pour soumettre et assembler par enchaînement les données du message provenant du client et les données externes cherchées de l'URL fourni. Les deux paragraphes suivants donnent les instructions pas à pas sur la façon de réaliser la "transmission sans téléchargement".

#### 8.4.1 Assemblage de message en utilisant l'extension IMAP CATENATE

Dans les variantes [RFC4468]/[RFC4469] de la stratégie de "transmission sans téléchargement" Lemonade, les messages sont initialement composés et édités dans un MUA. L'extension de la [RFC4469] à la [RFC3501] est alors utilisée pour créer les messages sur le serveur IMAP en transmettant un nouveau texte et en les assemblant. L'extension IMAP UIDPLUS [RFC4315] est utilisée par le client afin d'apprendre l'UID des messages créés. Finalement, un format d'URL [RFC4467] est donné à un serveur [RFC4409] pour une soumission utilisant l'extension BURL [RFC4468].

Le flux impliqué pour prendre en charge ce cas d'utilisation consiste en :

M : {à I -- Facultatif} Le client se connecte au serveur IMAP, commence facultativement TLS (si la confidentialité des données est requise) s'authentifie, ouvre une boîte aux lettres ("INBOX" dans l'exemple ci-dessous) et va chercher les structures de corps (voir la [RFC3501]).

Exemple :

M : A0051 UID FETCH 25627 (UID BODYSTRUCTURE)
I : * 161 FETCH (UID 25627 BODYSTRUCTURE (("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT"
1152 23)("TEXT" "PLAIN" ("CHARSET" "US-ASCII" "NAME" "trip.txt")
"<960723163407.20117h@washington.exemple.com>"
"Vos détails de voyage" "BASE64" 4554 73) "MIXED"))
I : A0051 OK terminé

M : {à I} Le client invoque CATENATE (voir la [RFC4469] pour les détails de sémantique et les étapes) -- cela permet au MUA de créer des messages sur le serveur IMAP en utilisant de nouvelles données combinées avec une ou plusieurs parties de message déjà présentes sur le serveur IMAP.

Noter que l'exemple pour cette étape n'utilise pas l'extension LITERAL+ de la [RFC2088]. Sans LITERAL+, le nouveau message est construit en utilisant trois allers-retours. Si LITERAL+ est utilisé, le nouveau message peut être construit en utilisant un aller retout.

M : A0052 APPEND Sent FLAGS (\Draft \Seen $MDNSent) CATENATE (TEXT {475}
I : + Prêt pour les données littérales
M : Message-ID: <419399E1.6000505@caernarfon.exemple.org>
M : Date: Thu, 12 Nov 2004 16:57:05 +0000
M : From: Bob Ar <bar@exemple.org>
M : MIME-Version: 1.0
M : To: foo@exemple.net
M : Subject: Sur votre voyage de vacances
M : Content-Type: multipart/mixed;
M : boundary="----------030308070208000400050907"
M :
M : ----------030308070208000400050907
M : Content-Type: text/plain; format=flowed
M :
M : Votre agent de voyage a envoyé le programme mis à jour.
M :
M : Salutations
M : Bob
M: ----------030308070208000400050907
M: URL "/INBOX;UIDVALIDITY=385759045/;
UID=25627/;Section=2.MIME" URL "/INBOX;
UIDVALIDITY=385759045/;UID=25627/;Section=2" TEXT {44}
I : + Prêt pour les données littérales
M :
M : ----------030308070208000400050907--
M : )
I : A0052 OK [APPENDUID 387899045 45] CATENATE Completed

M : {pour I} Le client utilise la commande GENURLAUTH pour demander un URL URLAUTH (voir la [RFC4467]).
I : {pour M} Le serveur IMAP retourne un URL URLAUTH convenable pour une retitution ultérieure avec URLFETCH (voir dans la [RFC4467] les détails de la sémantique et des étapes).

M : A0053 GENURLAUTH "imap://bob.ar@exemple.org/Sent;
UIDVALIDITY=387899045/;uid=45;expire=2005-10-28T23:59:59Z;urlauth=submit+bob.ar" INTERNAL
I : * GENURLAUTH "imap://bob.ar@exemple.org/Sent;
UIDVALIDITY=387899045/;uid=45;expire=2005-10-28T23:59:59Z;urlauth=submit+bob.ar:
internal:91354a473744909de610943775f92038"
I : A0053 OK GENURLAUTH terminé

M : {pour S} Le client se connecte au serveur de soumission de messagerie et commence une nouvelle transaction de messagerie. Il utilise BURL pour laisser le serveur de soumission SMTP aller chercher le contenu du message chez le serveur IMAP (voir la [RFC4467] pour les détails de la sémantique et des étapes -- cela permet au MUA d'autoriser le serveur de soumission SMTP à accéder au message composé par suite de l'étape CATENATE). Noter que la seconde commande EHLO est requise après une commande STARTTLS réussie. Noter aussi qu'il pourrait y avoir une troisième commande EHLO nécessaire si la seconde réponse EHLO ne mentionne aucune option BURL. Le paragraphe 8.4.2 montre cela.

S : 220 owlry.exemple.org ESMTP
M : EHLO potter.exemple.org
S : 250-owlry.exemple.com
S : 250-8BITMIME
S : 250-BINARYMIME
S : 250-PIPELINING
S : 250-BURL imap
S : 250-CHUNKING
S : 250-AUTH PLAIN
S : 250-DSN
S : 250-SIZE 10240000
S : 250-STARTTLS
S : 250 ENHANCEDSTATUSCODES
M : STARTTLS
S : 220 Prêt à commencer TLS
...négociation TLS, les données suivantes sont chiffrées...
M : EHLO potter.exemple.org
S : 250-owlry.exemple.com
S : 250-8BITMIME
S : 250-BINARYMIME
S : 250-PIPELINING
S : 250-BURL imap
S : 250-CHUNKING
S : 250-AUTH PLAIN
S : 250-DSN
S : 250-SIZE 10240000
S : 250 ENHANCEDSTATUSCODES
M : AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
M : MAIL FROM:<bob.ar@exemple.org>
M : RCPT TO:<foo@exemple.net>
S : 235 2.7.0 PLAIN authentification réussie.
S : 250 2.5.0 Address Ok.
S : 250 2.1.5 foo@exemple.net OK.
M : BURL imap://bob.ar@exemple.org/Sent;UIDVALIDITY=387899045/;
uid=45/;urlauth=submit+bar:internal:
91354a473744909de610943775f92038 LAST

S : {pour I} Le serveur de soumission de messagerie utilise URLFETCH pour aller chercher le message à envoyer. (Voir dans la [RFC4467] les détails de la sémantiques et des étapes. Le mécanisme d'autorisation dit "pawn-ticket" utilise un URI qui contient ses propres accréditifs d'autorisation.)

I : {pour S} Fournit le message composé à la suite de l'étape CATENATE).

Le serveur de soumission de messagerie ouvre une connexion IMAP avec le serveur IMAP :

I : * OK [CAPABILITY IMAP4REV1 STARTTLS NAMESPACE LITERAL+
CATENATE URLAUTH UIDPLUS CONDSTORE IDLE] imap.exemple.com
serveur IMAP prêt
S : a000 STARTTLS
I : a000 Commencer maintenant la négociation TLS
...négociation TLS, si elle réussit - les données suivantes sont chiffrées...
S : a001 LOGIN submitserver secret
I : a001 OK submitserver logged in
S : a002 URLFETCH "imap://bob.ar@exemple.org/Sent;
UIDVALIDITY=387899045/;uid=45/;urlauth=submit+bob.ar:
internal:91354a473744909de610943775f92038"
I : * URLFETCH "imap://bob.ar@exemple.org/Sent;
UIDVALIDITY=387899045/;uid=45/;urlauth=submit+bob.ar:
internal:91354a473744909de610943775f92038" {15065}
...le corps de message suit...

I : a002 OK URLFETCH terminé
S : a003 LOGOUT
I : * BYE À plus tard
I : a003 OK Déconnexion réussie

Noter que si la confidentialité des données n'est pas requise, le serveur de soumission de messagerie peut omettre la commande STARTTLS avant de produire la commande LOGIN.

S : {pour M} Le serveur de soumission assemble le message complet ; si l'assemblage réussit, il retourne OK au MUA:

S : 250 2.5.0 Ok.

M : {pour I} Le client marque le message contenant la pièce jointe transmise sur le serveur IMAP.

M : A0054 UID STORE 25627 +FLAGS.SILENT ($Forwarded)
I : * 215 FETCH (UID 25627 MODSEQ (12121231000))
I : A0054 OK STORE terminé

Note : la commande UID STORE montrée ci-dessus va seulement fonctionner si le message marqué est dans la boîte aux lettres actuellement choisie ; autrement, elle exige un SELECT. Cette commande peut être omise, car elle change simplement des méta données non opérationnelles non essentielles pour les opérations du client ou l'interopérabilité. La réponse FETCH non étiquetée est due à la [RFC4551]. Le mot-clé IMAP $Forwarded est décrit au paragraphe 5.9.

### 8.4.2 Assemblage de message en utilisant les extensions SMTP CHUNKING et BURL

Dans la variante [RFC4467]/[RFC4468] de la stratégie de "transmission sans téléchargement" Lemonade, les messages sont initialement composés et édités dans un MUA. Durant la soumission [RFC4409], les commandes BURL [RFC4468] et BDAT [RFC3030] sont utilisées pour créer des messages provenant de plusieurs parties. Les nouvelles parties de corps sont fournies en utilisant des commandes BDAT, tandis que les parties de corps existantes sont référencées en utilisant les URL de format de la [RFC4467] dans les commandes BURL.

Le flux impliqué pour prendre en charge ce cas d'utilisation consiste en : M : {pour I -- facultatif} Le client se connecte au serveur IMAP, commence facultativement TLS (si la confidentialité des données est requise) s'authentifie, ouvre une boîte aux lettres ("INBOX" dans l'exemple ci-dessous) et va chercher les structures de corps (voir la [RFC3501]).

Exemple :

M : B0051 UID FETCH 25627 (UID BODYSTRUCTURE)
I : * 161 FETCH (UID 25627 BODYSTRUCTURE (("TEXT" "PLAIN"
("CHARSET" "US-ASCII") NIL NIL "7BIT" 1152 23)(
"TEXT" "PLAIN" ("CHARSET" "US-ASCII" "NAME" "trip.txt")
"<960723163407.20117h@washington.exemple.com>"
"Détails de votre voyage" "BASE64" 4554 73) "MIXED"))
I : B0051 OK terminé

M : {pour I} Le client utilise la commande GENURLAUTH pour demander les URL URLAUTH (voir la [RFC4467]) en faisant référence aux pièces du message à assembler.
I : {pour M} Le serveur IMAP retourne les URL URLAUTH convenables pour une restitution ultérieure avec URLFETCH (voir dans la [RFC4467] les détails de la sémantique et des étapes).

M : B0052 GENURLAUTH "imap://bob.ar@exemple.org/INBOX;
UIDVALIDITY=385759045/;UID=25627/;Section=2.MIME;
expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar"
INTERNAL "imap://bob.ar@exemple.org/INBOX;
UIDVALIDITY=385759045/;UID=25627/;Section=2;
expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar" INTERNAL
I : * GENURLAUTH "imap://bob.ar@exemple.org/INBOX;
UIDVALIDITY=385759045/;UID=25627/;Section=2.MIME;
expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:

internal:A0DEAD473744909de610943775f9BEEF"
"imap://bob.ar@exemple.org/INBOX;
UIDVALIDITY=385759045/;UID=25627/;Section=2;
expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
internal:BEEFA0DEAD473744909de610943775f9"
I : B0052 OK GENURLAUTH terminé

M : {pour S} Le client se connecte au serveur de soumission de messagerie et commence une nouvelle transaction de messagerie. Il utilise BURL pour dire au serveur de soumission SMTP d'aller chercher sur le serveur IMAP les pièces de message à envoyer (voir dans la [RFC4468] les détails de la sémantique et des étapes).

Noter que la seconde commande EHLO est exigée après une commande STARTTLS réussie. La troisième commande EHLO est exigée si et seulement si la seconde réponse EHLO ne mentionne aucune option BURL.Voir au paragraphe 8.4.1 un exemple de soumission où la troisième commande/réponse EHLO n'est pas présente.

S : 220 owlry.exemple.org ESMTP
M : EHLO potter.exemple.org
S : 250-owlry.exemple.com
S : 250-8BITMIME
S : 250-BINARYMIME
S : 250-PIPELINING
S : 250-BURL
S : 250-CHUNKING
S : 250-AUTH DIGEST-MD5
S : 250-DSN
S : 250-SIZE 10240000
S : 250-STARTTLS
S : 250 ENHANCEDSTATUSCODES
M : STARTTLS
S : 220 Prêt à commencer TLS
...négociation TLS, les données qui suivent sont chiffrées...
M : EHLO potter.exemple.org
S : 250-owlry.exemple.com
S : 250-8BITMIME
S : 250-BINARYMIME
S : 250-PIPELINING
S : 250-BURL
S : 250-CHUNKING
S : 250-AUTH DIGEST-MD5 CRAM-MD5 PLAIN EXTERNAL
S : 250-DSN
S : 250-SIZE 10240000
S : 250 ENHANCEDSTATUSCODES
M : AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
S : 235 2.7.0 PLAIN authentification réussie.
M : EHLO potter.exemple.org
S : 250-owlry.exemple.com
S : 250-8BITMIME
S : 250-BINARYMIME
S : 250-PIPELINING
S : 250-BURL imap imap://imap.exemple.org
S : 250-CHUNKING
S : 250-AUTH DIGEST-MD5 CRAM-MD5 PLAIN EXTERNAL
S : 250-DSN
S : 250-SIZE 10240000
S : 250 ENHANCEDSTATUSCODES
M : MAIL FROM:<bob.ar@exemple.org> BODY=BINARY
S : 250 2.5.0 Address Ok.
M : RCPT TO:<foo@exemple.net>
S : 250 2.1.5 foo@exemple.net OK.
M : BDAT 475
M : Message-ID: <419399E1.6000505@caernarfon.exemple.org>

```
M : Date: Thu, 12 Nov 2004 16:57:05 +0000
M : From: Bob Ar <bar@exemple.org>
M : MIME-Version: 1.0
M : To: foo@exemple.net
M : Subject: À propos de votre voyage de vacances.
M : Content-Type: multipart/mixed;
M :     boundary="------------030308070208000400050907"
M :
M : --------------030308070208000400050907
M : Content-Type: text/plain; format=flowed
M :
M : Notre agent de voyage a envoyé le programme mis à jour.
M :
M : Salutations,
M : Bob
M : --------------030308070208000400050907
S : 250 2.5.0 OK
M : BURL imap://bob.ar@exemple.org/INBOX;
    UIDVALIDITY=385759045/;UID=25627/;Section=2.MIME;
    expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
    internal:A0DEAD473744909de610943775f9BEEF
S : 250 2.5.0 OK
M : BURL imap://bob.ar@exemple.org/INBOX;
     UIDVALIDITY=385759045/;UID=25627/;Section=2;
     expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
     internal:BEEFA0DEAD473744909de610943775f9
S : 250 2.5.0 OK
M : BDAT 44 LAST
M :
M : --------------030308070208000400050907--
```

S : {pour I} Le serveur de soumission de messagerie utilise URLFETCH pour aller chercher les pièces du message à envoyer. (Voir dans la [RFC4468] les détails de la sémantique et des étapes. Le mécanisme d'autorisation dit "pawn-ticket" utilise un URI qui contient ses propres accréditifs d'autorisation.)
I : {pour S} Retourne les parties de corps demandées.

Le serveur de soumission de messagerie ouvre une connexion IMAP avec le serveur IMAP :

```
I : * OK [CAPABILITY IMAP4REV1 STARTTLS NAMESPACE LITERAL+
         CATENATE URLAUTH UIDPLUS CONDSTORE IDLE] imap.exemple.com
         serveur IMAP prêt
S : b000 STARTTLS
I : b000 Commencer maintenant la négociation TLS
      ...négociation TLS, si elle réussit - les données suivantes sont chiffrées...
S : b001 LOGIN submitserver secret
I : b001 OK submitserver logged in
S : b002 URLFETCH "imap://bob.ar@exemple.org/INBOX;
         UIDVALIDITY=385759045/;UID=25627/;Section=2.MIME;
         expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
         internal:A0DEAD473744909de610943775f9BEEF" "imap://
         bob.ar@exemple.org/INBOX;
         UIDVALIDITY=385759045/;UID=25627/;Section=2;
         expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
         internal:BEEFA0DEAD473744909de610943775f9"
I : * URLFETCH "imap://bob.ar@exemple.org/INBOX;
         UIDVALIDITY=385759045/;UID=25627/;Section=2.MIME;
         expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
         internal:A0DEAD473744909de610943775f9BEEF" {84}
       ...une section de message suit...
          "imap://bob.ar@exemple.org/INBOX;
         UIDVALIDITY=385759045/;UID=25627/;Section=2;
```

expire=2006-10-28T23:59:59Z;urlauth=submit+bob.ar:
internal:BEEFA0DEAD473744909de610943775f9" {15065}
...une section de message suit...
I : b002 OK URLFETCH terminé
S : b003 LOGOUT
I : * BYE À plus tard
I : b003 OK Déconnexion réussie

Noter que si la confidentialité des données n'est pas requise, le serveur de soumission de messagerie peut omettre la commande STARTTLS avant de produire la commande LOGIN.

S : {pour M} Le serveur de soumission assemble le message complet ; si l'assemblage réussit, il accuse réception de l'acceptation du message par l'envoi d'une réponse 250 à la dernière commande BDAT :

S : 250 2.5.0 Ok, message accepté.

M : {pour I} Le client marque le message contenant la pièce jointe transmise sur le serveur IMAP.

M : B0053 UID STORE 25627 +FLAGS.SILENT ($Forwarded)
I : * 215 FETCH (UID 25627 MODSEQ (12121231000))
I : B0053 OK STORE terminé

Note : la commande UID STORE montrée ci-dessus va seulement fonctionner si le message marqué est dans la boîte aux lettres actuellement choisie ; autrement, cela exige un SELECT. Comme dans l'exemple précédent, cette commande n'est pas critique, et peut être omise. La réponse FETCH non étiquetée est due à la [RFC4551]. Le mot-clé IMAP $Forwarded est décrit au paragraphe 5.9.

### 8.5 Considérations sur la sécurité des pawn-tickets

Le mécanisme d'autorisation dit "pawn-ticket" utilise un URI, qui contient ses propres accréditifs d'autorisation en utilisant la [RFC4467]. L'avantage de ce mécanisme est que le serveur de soumission SMTP [RFC4409] ne peut pas accéder à des données sur le serveur [RFC4467] sans un "pawn-ticket" créé par le client.

Le "pawn-ticket" accorde l'accès seulement aux données spécifiques que le serveur de soumission SMTP [RFC4409] est autorisé à accéder, peut être révoqué par le client, et peut avoir une durée de validité limitée.

,

### 8.6 Copies des messages envoyés : le problème de fcc

Le "problème du fcc", ou "file carbon copy" se réfère à la délivrance d'une copie d'un message à une boîte aux lettres. De loin, le cas le plus commun de fcc est celui d'un client qui laisse une copie des messages sortants dans une boîte aux lettres "Messages envoyés" ou "Boîte de sortie".

Dans la stratégie traditionnelle, le MUA duplique les efforts de transmission au MSA en écrivant le message pour la destination fcc dans une étape séparée. Cela peut être une écriture sur un fichier de disque local ou un APPEND à une boîte aux lettres sur un serveur IMAP. Ce dernier est une des "transmissions de données répétitives du réseau" qui représente l'aspect "problème" du "problème du fcc".

L'extension BURL [RFC4468] peut être utilisée pour éliminer la transmission supplémentaire. Le message final est téléchargé à la boîte aux lettres désignée pour la messagerie sortante par la commande APPEND de la [RFC3501]. Noter que quand il fait ainsi, le client devrait utiliser les mots-clés $SubmitPending et $Submitted IMAP décrits au paragraphe 5.10. Noter aussi que APPEND, y compris quand amélioré par la [RFC4469], peut seulement créer une seule copie du message et c'est seulement utilisé sur le serveur qui reçoit le message sortant pour soumission. Des copies supplémentaires du message sur le même serveur peuvent être créées par en utilisant une ou plusieurs commandes COPY.

## 9. Considérations de déploiement

Les considérations de déploiement sont discutées extensivement dans la [RFC5383].

## 10. Considérations sur la sécurité

Il est conseillé aux mises en œuvre d'examiner les considérations sur la sécurité de tous les documents référencés. Cette section les souligne simplement, et conseille les mises en œuvre sur des questions spécifiques relatives à la combinaison des extensions.

Les considérations sur la sécurité de la "transmission sans téléchargement" Lemonade sont discutées dans la Section 8. Des considérations sur la sécurité supplémentaires se trouvent dans les [RFC3501], [RFC4409], [RFC5228], et autres documents qui décrivent d'autres extensions SMTP, IMAP, et Sieve composant le profil Lemonade.

Noter que le mécanisme d'authentification de mise en œuvre obligatoire pour la soumission SMTP est décrit dans la [RFC4954]. Le mécanisme d'authentification de mise en œuvre obligatoire pour IMAP est décrit dans la [RFC3501].

### 10.1 Protection de la confidentialité des messages soumis

Quand les clients soumettent de nouveaux messages, la protection de liaison comme celle de la [RFC5246] garde contre un espion qui voit le contenu du message soumis. On notera cependant que même si TLS n'est pas utilisé, les risques pour la sécurité ne sont pas pire si BURL est utilisé pour référencer le texte que si le texte est soumis directement. Si BURL n'est pas utilisé, un espion obtient l'accès à tout le texte du message. Si BURL est utilisé, l'espion peut ou non être capable d'obtenir un tel accès, selon la forme de BURL utilisée. Par exemple, certaines formes restreignent l'usage de l'URL à une entité autorisée comme un serveur de soumission ou un utilisateur spécifique.

### 10.2 TLS

Quand les clients Lemonade utilisent l'extension BURL pour la soumission de messagerie, une extension qui exige d'envoyer un jeton URLAUTH au serveur de soumission de messagerie, ce jeton devrait être protégé de l'interception pour éviter une attaque en répétition qui peut divulguer le contenu du message à un attaquant. Le chiffrement fondé sur la [RFC5246] de la session IMAP qui produit le GENURLAUTH et le chemin de soumission de messagerie va fournir une protection contre cette attaque.

Les serveurs de soumission de messagerie conformes à Lemonade DEVRAIENT utiliser des connexions IMAP protégées par TLS quand ils vont chercher le contenu du message en utilisant le jeton URLAUTH fourni par le client Lemonade.

Quand un client utilise SMTP STARTTLS pour envoyer une commande BURL qui fait référence à des informations non publiques, l'utilisateur s'attend à ce que le contenu entier du message soit traité de façon confidentielle. Pour satisfaire cette attente, le serveur de soumission de message DEVRAIT utiliser STARTTLS ou un mécanisme fournissant une confidentialité équivalente des données quand il va chercher le contenu référencé par cet URL.

### 10.3 Extensions supplémentaires et modèles de déploiement

La présente spécification ne fournit pas de mesures de sécurité supplémentaires au delà de celle des documents référencés de messagerie Internet et de Lemonade.

On note cependant les risques pour la sécurité associés à :
- o la notification hors bande,
- o la configuration de serveur par le client,
- o La configuration de client par le serveur,
- o la présence de serveurs mandataires,
- o la présence de serveurs comme intermédiaires,
- o en général, les modèles de déploiement considérés par OMA MEM qui ne sont pas conventionnels pour l'IETF,
- o des mesures pour traiter un besoin perçu de traverser des pare-feu et des intermédiaires de réseau mobile.

Les déploiements qui fournissent ces services supplémentaires ou opèrent dans ces environnements doivent consulter les considérations de sécurité pour les normes et les pratiques de sécurité organisationnelles pertinentes.

## 11. Considérations relatives à l'IANA

Les capacités IMAP4 sont enregistrées par revue de l'IETF, comme défini dans la [RFC5226]. Le présent document définit la capacité URL-PARTIAL IMAP (paragraphe 5.7.1). L'IANA a ajouté cette extension au registre "IMAP Capability".

## 12. Changements par rapport à la RFC 4550

Quand on le compare à la RFC 4550, le présent document ajoute les exigences supplémentaires suivantes pour un serveur IMAP conforme à Lemonade :

Extensions IMAP : BINARY, COMPRESS=DEFLATE, CONTEXT=SEARCH, CONTEXT=SORT, CONVERT, ENABLE, ESEARCH, ESORT, I18NLEVEL=1, NOTIFY, QRESYNC, SASL-IR, SORT, URL-PARTIAL;

Mots clés IMAP : $SubmitPending, $Submitted.

Autres exigences : exige que tout serveur IMAP conforme à Lemonade prenne en charge le code de réponse CAPABILITY.

Quand on le compare à la RFC 4550, le présent document ajoute les nouvelles exigences suivantes pour un agent de livraison de message conforme à Lemonade :

Prise en charge du langage de filtrage Sieve, avec les extensions Sieve suivantes : ENOTIFY, IMAP4FLAGS, RELATIONAL, VACATION, VARIABLES, comparator-i;unicode-casemap.

Quand on le compare à la RFC 4550, le présent document recommande l'utilisation de la méthode de compression DEFLATE pour TLS. Toutes les autres exigences restent les mêmes.

De plus, les changements/améliorations suivantes ont été faites à la RFC 4550 (la liste pourrait être incomplète) :

Un nouveau paragraphe avec des exigences supplémentaires pour les agents d'utilisateur de messagerie Lemonade a été ajouté, en particulier ils sont obligés de prendre en charge le paramètre Format=flowed au type de support text/plain.

L'usage du mot-clé IMAP $Forwarded a été précisé.

Les exemples de transmission sans téléchargement ont été corrigés et étendus.

Ajout d'un nouveau paragraphe décrivant les notifications dans la bande et hors bande provenant d'un magasin de messagerie conforme à Lemonade.

## 13. Remerciements

Les éditeurs remercient et ont apprécié le travail et les commentaires du groupe de travail Lemonade de l'IETF et du groupe de travail OMA MEM.

En particulier, les éditeurs tiennent à remercier Eric Burger, Glenn Parsons, Randall Gellens, Filip Navara, Zoltan Ordogh, Greg Vaudreuil, et Fan Xiaohui de leurs commentaires et de leur relecture.

## 14. Références

### 14.1 Références normatives

[RFC1652] J. Klensin et autres, "Extensions de service SMTP pour transport MIME sur 8 bits", juillet 1994. *(Remplacée par* RFC6152) *(D.S.)*

[RFC1870] J. Klensin, N. Freed, K. Moore, "Extensions de service à SMTP pour déclaration de taille de message",

novembre 1995. (STD0010)

[RFC2034] N. Freed, "Extension de service SMTP pour le retour de codes d'erreur améliorés", octobre 1996. *(P.S.)*

[RFC2088] J. Myers, "Littéraux IMAP4 sans synchronisation", janvier 1997. (*P.S. . MàJ par* RFC4466 . *Remplacée par* RFC7888)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. DOI 10.17487/RFC2119, *(MàJ par* RFC8174)

[RFC2177] B. Leiba, "Commande IDLE dans IMAP4", juin 1997. *(P.S.)*

[RFC2342] M. Gahrns, C. Newman, "Espace de noms IMAP4", mai 1998. (*MàJ par* RFC4466) *(P.S.)*

[RFC2920] N. Freed, "Extension de service SMTP pour le traitement de commandes en parallèle", septembre 2000. (STD0060)

[RFC3030] G. Vaudreuil, "Extensions de service SMTP pour la transmission de grands messages MIME binaires", décembre 2000. *(P.S.)*

[RFC3207] P. Hoffman, "Extension de service SMTP pour un SMTP sécurisé sur TLS", février 2002. *(P.S., MàJ par* RFC7817)

[RFC3461] K. Moore, "Extension de service du protocole simple de transfert de messagerie (SMTP) pour les notifications d'état de livraison (DSN)", janvier 2003. (*MàJ par* RFC3798, 885, 5337, 6533, 8098) *(D.S.)*

[RFC3501] M. Crispin, "Protocole d'accès au message Internet - version 4rev1", mars 2003. *(P.S. ; MàJ par* RFC4466, 4469, 4551, 5032, 5182, 7817, 8314, 8437, 8474 ; *remplacée par la* RFC9051

[RFC3516] L. Nerenberg, "Extension Contenu binaire à IMAP4", avril 2003. (*MàJ par* RFC4466) *(P.S.)*

[RFC3676] R. Gellens, "Format Text/Plain et paramètres DelSp", février 2004. (*Remplace* RFC2646) *(P.S.)*

[RFC3749] S. Hollenbeck, "Méthodes de compression du protocole de sécurité de la couche transport", mai 2004. *(P.S.)*

[RFC4315] M. Crispin, "Protocole d'accès au message Internet (IMAP) - extension UIDPLUS", décembre 2005. *(P.S.)*

[RFC4409] R. Gellens, J. Klensin, "Soumission du message de messagerie électronique", avril 2006. (*Remplacé par la* RFC6409 STD072*)*

[RFC4467] M. Crispin, "Protocole d'accès au message Internet (IMAP) - Extension URLAUTH", mai 2006. (*P.S. ; MàJ par* RFC5092)

[RFC4468] C. Newman, "Extension BURL de soumission de message", mai 2006. (*MàJ* RFC3463) (*MàJ par* RFC5248) *(P.S.)*

[RFC4469] P. Resnick, "Extension CATENATE au protocole d'accès au message Internet (IMAP)", avril 2006. *(P.S.)*

[RFC4551] A. Melnikov, S. Hole, "Extension à IMAP pour l'opération STORE conditionnelle ou la resynchronisation des changements rapides de fanion", juin 2006. (*MàJ* RFC3501) *(P.S. ; Remplacée par RFC*7162*)*

[RFC4731] A. Melnikov, D. Cridland, "Extension de commande SEARCH à IMAP4 pour contrôler le type d'information retournée", novembre 2006. *(P.S.)*

[RFC4954] R. Siemborski et A. Melnikov, éd., "Extension de service à SMTP pour l'authentification", juillet 2007. (*Remplace* RFC2554) (*MàJ* RFC3463) (*MàJ par* RFC5248) *(P.S.)*

[RFC4959] R. Siemborski, A. Gulbrandsen, "Extension à IMAP pour la réponse de client initial d'authentification simple et couche de sécurité (SASL)", septembre 2007. *(P.S.)*

[RFC4978] A. Gulbrandsen, "Extension COMPRESS à IMAP", août 2007. *(P.S.)*

[RFC5051] M. Crispin, "i;unicode-casemap : un algorithme simple d'interclassement pour chaînes Unicode", octobre 2007. *(P.S.)*

[RFC5092] A. Melnikov et C. Newman, "Schéma d'URL IMAP", novembre 2007. (*Remplace* RFC2192 ; *MàJ* RFC4467 ; *MàJ par* RFC5593) *(P.S.)*

[RFC5161] A. Gulbrandsen et A. Melnikov,"Extension ENABLE à IMAP", mars 2008. *(P.S.)*

[RFC5162] A. Melnikov et autres, "Extensions à IMAP4 pour la resynchronisation rapide de boîte aux lettres", mars 2008. *(P.S.; obsolète, voir RFC7162)*

[RFC5228] P. Guenther et autres, "Sieve : un langage de filtrage de messagerie électronique", janvier 2008. (*P.S. ; Remplace* RFC3028, *MàJ par* RFC5229, 5429 9042*)*

[RFC5229] K. Homme, "Filtrage de messagerie Sieve : extension Variables", janvier 2008. (*MàJ par* RFC5173) *(P.S.)*

[RFC5230] T. Showalter, N. Freed, éd., "Filtrage de messagerie Sieve : extension Vacation", janvier 2008. *(P.S. ; MàJ par* RFC8580*.)*

[RFC5231] W. Segmuller, B. Leiba, "Filtrage de messagerie Sieve : extension Relational", janvier 2008. (*Remplace* RFC3431*)(P.S.)*

[RFC5232] A. Melnikov, "Filtrage de messagerie Sieve : extension Imap4flags", janvier 2008. *(P.S.)*

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", DOI 10.17487/RFC5246, août 2008. *(P.S. ; remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 ; *rendue obsolète par la* RFC8446)

[RFC5255] C. Newman et autres, "Internationalisation du protocole d'accès au message Internet", juin 2008. *(P.S.)*

[RFC5256] M. Crispin, K. Murchison, "Protocole d'accès au message Internet - extensions SORT et THREAD", juin 2008. (*MàJ par* RFC5957) *(P.S.)*

[RFC5259] A. Melnikov et P. Coates, "Protocole d'accès au message Internet - extension CONVERT", juillet 2008. *(P.S.)*

[RFC5267] D. Cridland, C. King, "Contextes pour IMAP4", juillet 2008. (*MàJ par* RFC5465) *(P.S.)*

[RFC5435] A. Melnikov et autres, "Filtrage de messagerie Sieve : extension pour les notifications", janvier 2009. *(P.S. ; MàJ par* RFC8580*)*

[RFC5465] A. Gulbrandsen, C. King, A. Melnikov, "Extension IMAP NOTIFY", février 2009. (*MàJ* RFC5267) *(P.S.)*

### 14.2 Références pour information

[Err1807] RFC Errata, Errata ID 1807, RFC 5162, <http://www.rfc-editor.org>.

[Err1808] RFC Errata, Errata ID 1808, RFC 5162, <http://www.rfc-editor.org>.

[Err1809] RFC Errata, Errata ID 1809, RFC 5162, <http://www.rfc-editor.org>.

[Err1810] RFC Errata, Errata ID 1810, RFC 5162, <http://www.rfc-editor.org>.

[OMA-EMN] Open Mobile Alliance, "Open Mobile Alliance Email Notification Version 1.0", OMA http://www.openmobilealliance.org/Technical/release_program/emn_v10.aspx, octobre 2007.

[OMA-MEM-ARCH] Open Mobile Alliance, "Mobile Email Architecture Document", OMA (Work in Progress), http://www.openmobilealliance.org/, octobre 2005.

[OMA-MEM-REQ] Open Mobile Alliance, "Mobile Email Requirements Document", OMA http://www.openmobilealliance.org/release_program/docs/RD/OMA-RD-MobileEmail-V1_0_20051018-C.pdf, oct 2005.

[RFC4146] R. Gellens, "Notification simple de nouveau message", août 2005. *(Information)*

[RFC4549] A. Melnikov, éd., "Opérations de synchronisation pour clients IMAP4 déconnectés", juin 2006. *(Information)*

[RFC5068] C. Hutzler et autres, "Opérations de soumission de message électronique : exigences d'accès et de comptabilité", novembre 2007. (BCP0134) *(MàJ par* RFC8314)

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[RFC5321] J. Klensin, "Protocole simple de transfert de messagerie(SMTP)", octobre 2008. (*Remplace* RFC2821) (*MàJ* RFC1123) *(D.S.)*

[RFC5383] R. Gellens, "Considérations sur le déploiement de la messagerie mobile conforme à Lemonade", octobre 2008. (BCP0143)

[RFC5442] E. Burger, G. Parsons, "Architecture LEMONADE – prise en charge de la messagerie électronique mobile (MEM) de Open Mobile Alliance (OMA) avec la messagerie Internet" mars 2009. *(Information)*

[RFC5464] C. Daboo, "Extension IMAP METADATA", février 2009. *(P.S.)*

[RFC5466] A. Melnikov, C. King, "Extension IMAP4 pour les recherches désignées (filtres)", février 2009. *(P.S.)*

[RFC5551] R. Gellens, "Architecture des notifications Lemonade", août 2009. *(Info)*

[RFC5598] D. Crocker, "Architecture de la messagerie Internet", juillet 2009. *(Information)*

[RFC5804] A. Melnikov, T. Martin, "Protocole pour la gestion distante des scénarios Sieve", juillet 2010. *(P. S. ; MàJ par* 7817)

## Appendice A. Errata

Errata ID: 1807 [Err1807]
Statut : Vérifié
Type : Technique
Rapporté par : Timo Sirainen
Date de rapport : 2009-07-14
Nom du vérificateur : Alexey Melnikov
Date de vérification : 2009-07-18

La Section 1 dit :

Elle devrait dire :

Une fois qu'une commande activant "CONDSTORE" est produite par le client, le serveur DOIT automatiquement inclure les données d'UID et de "mod-sequence" dans toutes les réponses FETCH non étiquetées suivantes (jusqu'à ce que la connexion soit close) qu'elles aient été causées par un STORE/UID STORE régulier, un STORE/UID STORE avec un modificateur UNCHANGEDSINCE, ou un agent externe. Noter que cette règle n'affecte pas les réponses FETCH non étiquetées causées par une commande FETCH qui n'inclut pas d'élément de données d'UID et/ou de MODSEQ FETCH, ou d'UID FETCH sans l'élément de données MODSEQ FETCH.

Notes :
Raison :
Il est très difficile aux clients d'utiliser des réponses FETCH non sollicitées sans le champ UID. Cela est rendu encore pire par le texte qui dit "les serveurs NE DEVRAIENT PAS envoyer des UID pour les messages précédemment purgés [dans les réponses VANISHED]". Comme ce n'est pas un "NE DOIT PAS", une conversation avec un serveur conforme à la RFC

pourrait être par exemple :

A1 NOOP
* 0 EXISTS
A1 OK
A2 NOOP
* 10 EXISTS
* VANISHED 1000:2000
* 3 FETCH (FLAGS (\Seen) MODSEQ (14749))
* 5 FETCH (FLAGS (\Seen) MODSEQ (14749))
* VANISHED 2000:3000
A2 OK NOOP Completed

Le client ne pourrait rien faire avec les informations provenant des réponses FETCH parce que il ne peut savoir à quels messages elles se réfèrent.

Errata ID : 1808 [Err1808]
Statut : Vérifié
Type : Technique
Rapporté par : Timo Sirainen
Date de rapport : 2009-07-14
Nom du vérificateur : Alexey Melnikov
Date de vérification : 2009-07-18

Le paragraphe 3.4 dit :
Si au moins un message a été purgé, le serveur DOIT envoyer la séquence de modifications mises à jour par boîte aux lettres en utilisant le code de réponse HIGHESTMODSEQ (défini dans CONDSTORE) dans la réponse OK étiquetée.

Exemple :
C: A202 CLOSE
S: A202 OK [HIGHESTMODSEQ 20010715194045319] fait

Il devrait dire :
Le serveur NE DOIT PAS envoyer la séquence de modifications mises à jour par boîte aux lettres en utilisant le code de réponse HIGHESTMODSEQ (défini dans CONDSTORE) dans la réponse OK étiquetée, car cela pourrait causer la perte de synchronisation chez le client.

Exemple :
C: A202 CLOSE
S: A202 OK fait

Notes :
Raison :
Le HIGHESTMODSEQ ne peut pas être utilisé de façon fiable à moins que le serveur envoie au client tous les changements faits par les autres clients. Même alors il est difficile aux clients et aux serveurs de mettre cela en œuvre. Par exemple :

C1: 2 STORE 1 +FLAGS.SILENT \supprimé
1: * 1 FETCH (MODSEQ 1)
1: 2 OK

C2: 1 STORE 2 +FLAGS.SILENT \supprimé
S1: * 2 FETCH (MODSEQ 2)
S2: 1 OK

C1: 3 CLOSE
S1: 3 [HIGHESTMODSEQ 3]

Le client pensait probablement que seul le message 1 était purgé, donc il n'enregistre pas la seconde purge. Et il ne le fera probablement jamais si il utilise QRESYNC pour trouver seulement les nouvelles purges.

Un exemple encore pire serait si le second client avait aussi supprimé le fanion \Supprimé du message 1. Le premier client aurait alors enregistré le mauvais message à purger.

Errata ID : 1809 [Err1809]
Statut : Vérifié
Type : Technique
Rapporté par : Timo Sirainen
Date de rapport : 2009-07-14
Nom du vérificateur : Alexey Melnikov
Date de vérification : 2009-07-18

La Section 5 dit :

Après avoir achevé une synchronisation complète, le client DOIT aussi prendre note de tout élément de données MODSEQ FETCH non sollicité reçu du serveur. Chaque fois que le client reçoit une réponse étiquetée à une commande, il calcule la plus grande valeur parmi tous les éléments de données MODSEQ FETCH reçus depuis la dernière réponse étiquetée. Si cette valeur est supérieure à celle de la copie du client de la valeur de HIGHESTMODSEQ, alors le client DOIT utiliser cette valeur comme nouvelle valeur de HIGHESTMODSEQ.

Note : Il n'est pas sûr de mettre à jour la copie du client de la valeur HIGHESTMODSEQ avec une valeur d'élément de données MODSEQ FETCH aussitôt qu'elle est reçue parce que les serveurs ne sont pas obligés d'envoyer des éléments de données MODSEQ FETCH en ordre croissant de modseqence. Cela peut conduire à ce que le client manque des changements en cas de perte de connexité.

Elle devrait dire :

Après avoir achevé une synchronisation complète, le client DOIT aussi prendre note de tout élément de données non sollicité MODSEQ FETCH et des codes de réponse HIGHESTMODSEQ reçus du serveur. Chaque fois que le client reçoit une réponse étiquetée à une commande, il vérifie la réponse non sollicitée reçue pour calculer la nouvelle valeur de HIGHESTMODSEQ. Si le code de réponse HIGHESTMODSEQ est reçu, le client DOIT l'utiliser même si il a vu des mod-sequences supérieures. Autrement, le client calcule la plus forte valeur parmi tous les éléments de données MODSEQ FETCH reçus depuis la dernière réponse étiquetée. Si cette valeur est supérieure à la copie du client de la valeur de HIGHESTMODSEQ, alors le client DOIT utiliser cette valeur comme sa nouvelle valeur de HIGHESTMODSEQ.

Exemple :
C: A1 STORE 1:2 (UNCHANGEDSINCE 96) +FLAGS.SILENT \Seen
S: * 1 FETCH (UID 6 MODSEQ (103))
S: * 2 FETCH (UID 7 MODSEQ (101))
S: * OK [HIGHESTMODSEQ 99] La réponse VANISHED avec MODSEQ 100 est retardée
S: A1 OK [MODIFIED 3] fait

C: A2 STORE 3 +FLAGS.SILENT \Seen
S: * 3 FETCH (UID 8 MODSEQ (104))
S: A2 OK [HIGHESTMODSEQ 99] Retarde encore VANISHED

C: A3 NOOP
S: * VANISHED 8
S: A3 OK [HIGHESTMODSEQ 104] fait

Note : Il n'est pas sûr de mettre à jour la copie du client de la valeur HIGHESTMODSEQ avec une valeur d'élément de données MODSEQ FETCH aussitôt qu'elle est reçue parce que les serveurs ne sont pas obligés d'envoyer des éléments de données MODSEQ FETCH en ordre croissant de modseqence. Certaines commandes peuvent aussi retarder les réponses EXPUNGE (ou VANISHED) avec de plus petites mod-sequences. Cela peut conduire à ce que le client manque des changements en cas de perte de connexité.

Notes : raison: Autrement les clients pourraient perdre des changements en cas de perte de connexité.

Errata ID: 1810 [Err1810]

Statut : Vérifié
Type : Technique
Rapporté par : Timo Sirainen
Date de rapport : 2009-07-14
Nom du vérificateur : Alexey Melnikov
Date de vérification : 2009-07-18

La Section 1 dit :

Elle devrait dire :

Un serveur qui met en œuvre QRESYNC DOIT envoyer des événements non étiquetés au client d'une façon qui ne fasse pas perdre de changements au client en cas de perte de connexité. En particulier cela signifie que si serveur envoie des éléments de données MODSEQ FETCH alors que des réponses EXPUNGE (ou VANISHED) avec des mod-sequences inférieures sont retardées, le serveur DOIT envoyer un code de réponse HIGHESTMODSEQ avec une valeur inférieure au mod-sequence du EXPUNGE. Voir l'exemple de la Section 5.

Note : Ceci se rapporte à l'autre errata de la Section 5, qui décrit quel devrait être le comportement du client. Ceci décrit quel devrait être le comportement du serveur. Il aurait été bien de les mettre dans la même section, mais cela exigerait probablement de plus grands changements.

## Adresse des éditeurs

Dave Cridland
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK
mél : dave.cridland@isode.com

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK
mél : Alexey.Melnikov@isode.com

Stephane H. Maes
Oracle
MS 4op634, 500 Oracle Parkway
Redwood Shores, CA 94539
USA
téléphone : +1-203-300-7786
mél : stephane.maes@oracle.com