# RFC 5598 : Internet Mail Architecture

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juillet 2009

Date de publication du RFC : Juillet 2009

https://www.bortzmeyer.org/5598.html

---

Le courrier électronique est aujourd'hui une des principales applications de l'Internet. Mais, bien qu'il existe plusieurs RFC normalisant ses composantes techniques (comme les RFC 5321 [1] et RFC 5322), aucun document « officiel » ne décrivait l'**architecture** du courrier, ses principales composantes, leurs relations, et les termes à utiliser pour les désigner. C'est le rôle de ce nouveau RFC, dont le développement souvent douloureux a pris près de cinq ans.

Depuis bientôt trente ans qu'il existe, le courrier électronique a connu des hauts et des bas dans les médias. Très souvent, sa fin proche a été proclamée, au profit, par exemple, de la messagerie instantanée. On a même vu des discours à la limite du racisme, affirmant que le courrier électronique n'avait pas de sens en dehors de l'Europe parce que les africains seraient éternellement voués à la culture orale <https://www.bortzmeyer.org/l-afrique-et-l-ecrit.html>. Mais, malgré la concurrence de nouveaux gadgets qui brillent (le plus récent étant Google Wave), malgré les assauts du spam, malgré les réglements intérieurs décourageant l'usage du courrier électronique (comme je l'ai vu récemment dans une entreprise dont 100 % de l'activité concerne l'Internet), le traditionnel courrier marche encore très bien. Ce n'était donc pas inutile de documenter son architecture, tâche à laquelle s'est attelée un vétéran, Dave Crocker.

Le courrier s'étant développé de manière relativement informelle, sans architecture pré-définie (au contraire du défunt X.400, qui avait tenté une approche de haut en bas), même le vocabulaire n'est pas normalisé. La différence entre "*forwarding*" et "*redirecting*" n'a jamais ainsi été clairement expliquée, et c'est encore pire si on veut écrire en français. Résultat, les noms des menus dans les logiciels ne sont pas cohérents d'un logiciel à l'autre, et les discussions techniques sont souvent lentes et pénibles car il n'y a pas d'accord sur les concepts de base. C'est ainsi que le groupe de travail Marid <http://tools.ietf.org/wg/marid/> avait perdu beaucoup de temps dans des débats byzantins.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5321.txt

Le RFC 5598 va donc souvent créer un vocabulaire nouveau, qui déroutera tout le monde.

Autre problème lorsqu'il faut décrire l'architecture du courrier : il a beaucoup évolué depuis le début. Comme un service de messagerie sans utilisateurs ne sert à rien, chaque changement s'est fait en maintenant la compatibilité et le résultat n'est donc pas toujours très organisé.

Ce RFC 5598 ne vise pas à améliorer le courrier, uniquement à décrire comment il fonctionne **actuellement**. Il résume le fonctionnement du courrier (section 1). Celui-ci repose sur trois séries de normes :

— Le protocole SMTP (aujourd'hui normalisé dans le RFC 5321) qui décrit le moyen de faire circuler un message sur l'Internet,
— Le format des messages, dit souvent « RFC 822 » (aujourd'hui normalisé dans le RFC 5322) et que notre RFC nomme IMF ("*Internet Message Format*"),
— Une extension à ce format, MIME (aujourd'hui normalisé dans le RFC 2045), qui permet de gérer du texte dans différents jeux de caractères et des contenus multimédias.

La connaissance de l'histoire est souvent utile pour comprendre l'existant, d'autant plus que l'Internet repose largement sur le respect de la base installée : pas question de supprimer tout à coup un service utile. La section 1.1 retrace donc les (très) grandes lignes de l'évolution du courrier. Celui-ci a eu son premier document d'architecture avec le RFC 1506, qui empruntait à X.400 le vocabulaire de MUA et MTA. L'ensemble des MTA formait le **MHS** ("*Message Handling System*"). Mais les piliers que sont le protocole SMTP, le format IMF des messages, et le format des adresses avec le fameux @, sont restés très constants. Autre constante, la séparation entre le contenu du message et les informations de contrôle, qui permettent son acheminement.

Sont également restés les principes suivants :

— Adressage global (`stephane+blog@bortzmeyer.org` fonctionne partout et désigne toujours la même boîte aux lettres),
— Échange **asynchrone**, la grosse différence entre le courrier et la plupart des autres modes de communication,
— Absence d'accord préalable entre les acteurs. Un étudiant chilien peut écrire à un chercheur d'une entreprise indienne, sans qu'aucun contrat n'aie été signé, ou aucun contact pris précédemment. (C'est un des points les plus attaqués par ceux qui, comme le MAAWG, voudraient limiter le courrier à un cartel de gros opérateurs.) Mais le RFC note que ce point est **le** principal facteur de succès du courrier électronique et qu'il doit être maintenu.

Autre méta-question, quel est le rôle d'une architecture ? La section 1.2 la voit comme la connexion entre un **service** rendu à l'utilisateur et le(s) **protocole(s)** qui mettent en œuvre ce service. C'est l'architecture qui permet de s'y retrouver dans les protocoles utilisés. Elle est donc d'un plus haut niveau que les protocoles. Mais attention, le protocole ne respecte pas forcément complètement l'architecture (surtout comme lorsque, comme ici, il a été développé bien avant...) et ce n'est pas forcément un défaut, l'architecture doit être une aide, pas un réglement à respecter aveuglément.

Bien, maintenant que les bases sont posées, chaque section va parler d'une classe d'objets différente. La section 2 commence avec les **rôles des différents acteurs** ("*actor roles*"). Quels sont les rôles joués ici ?

D'abord, il y a les utilisateurs ("*user actors*"), expliqués en section 2.1. Ce ne sont pas forcément des humains, il existe aussi des programmes qui écrivent, traitent et lisent les messages. Il y a quatre sortes d'utilisateurs (le RFC contient un diagramme de leurs interactions, en page 10) :

— les auteurs,
— les destinataires,
— les répondeurs ("*return handlers*"),
— les intermédiaires ("*mediators*").

L'auteur est responsable du contenu du message, qu'il confie au MHS pour que celui-ci l'achemine au destinataire. Celui-ci lit le message. S'il y répond, créant un nouveau message, il devient Auteur et le précédent auteur devient Destinataire de ce nouveau message. Les répondeurs se chargent de générer des réponses lors d'évènements comme une adresse de destination inexistante (dans ce cas, l'auteur reçoit une réponse qui ne vient pas du destinataire).

Plus complexe, l'intermédiaire (section 2.1.4) va recevoir le message et le réémettre à des destinataires différents, parfois après modification. Un intermédiaire typique est un gestionnaire de liste de diffusion. Le RFC suggère une bonne définition pour un intermédiaire : il envoie des messages donc il est Auteur, mais les Destinataires de ces messages ne le considèrent pas comme tel.

Le second rôle est celui des serveurs ("*MHS actors*"), vus en section 2.2. Ce sont les programmes qui vont, collectivement, mettre en œuvre ce que les utilisateurs perçoivent comme une entité unique, le MHS ("*Message Handling System*"). La page 13 contient un joli diagramme de leurs interactions. Il y a :

— l'origine ("*originator*"), le premier serveur qui reçoit le message, et a notamment pour travail de s'assurer de sa validité,
— le relais ("*relay*") qui reçoit le message d'un serveur et le transmet à un autre (un message passe souvent par plusieurs relais, le courrier n'étant pas implémenté directement entre deux machines). Suivant le RFC 2505, le relais ajoutera une trace dans les en-têtes des messages (l'en-tête `Received:`). Le relais travaille à un niveau en dessous de l'Intermédiaire, mais un niveau au dessus des routeurs IP,
— le récepteur ("*receiver*"), le dernier serveur à traiter le message,
— la passerelle ("*gateway*"), un serveur un peu particulier, qui tient de l'Intermédiaire et du Relais (section 2.2.3). Son travail est de faire passer le courrier vers des mondes utilisant d'autres types de courrier, par exemple un autre format que l'IMF du RFC 5322.

Après les Utilisateurs et les Serveurs, un autre rôle important est celui des Organisations ("*administrative actors*", section 2.3). Plus formellement, on les nomme ADMD ("*ADministrative Management Domain*"). C'est au sein d'un ADMD que se prennent les décisions et que sont appliquées des politiques, par exemple des choix en terme de lutte anti-spam. Au niveau de l'ensemble de l'Internet, il n'y a pas unité de décision, il n'est pas envisageable de demander, par exemple, un déploiement généralisé de telle ou telle technique. En revanche, au sein d'un ADMD, de telles décisions sont possibles (c'est donc au courrier ce qu'un AS est au routage.) Beaucoup de décisions, par exemple en matière de filtrage, dépendent de si le message est interne à l'ADMD ou pas.

Le RFC 5598 discerne plusieurs sortes d'ADMD :

— Ceux du bord ("*Edge*"),
— Les consommateurs ("*Consumer*"),
— Ceux de transit ("*Transit*").

Un dessin page 17 illustre leurs relations. Lors de l'envoi d'un message, la transmission se fait en général entre les deux ADMD de bord, directement (à ce niveau d'analyse ; à un niveau plus bas, il y a plusieurs MTA). Parfois, un ou plusieurs ADMD de transit traitent le message. Parfois encore, l'ADMD du bord garde le message pour l'ADMD consommateur, comme lorsqu'il s'agit d'un accès Web aux boîtes aux lettres. (Voir aussi le RFC 5068.)

Pour désigner toutes les entités qui apparaissent dans le courrier, il faut des **identificateurs**. Il existe plusieurs identités possibles, exposées en section 3 :

— Les boîtes aux lettres, identifiées par une adresse de courrier comme `Jean.Durand@comptabilite.monentrepr`
La partie à gauche du @ doit être considérée comme **opaque** par la plupart des programmes qui manipulent le courrier, surtout lorsqu'ils n'ont pas lu la norme `<https://www.bortzmeyer.org/arreter-d-interdire-des-adresses-legales.html>`. Elle ne doit être interprétée qu'à la destination finale. Les conventions comme les adresses incluant un + sont purement locales et ne doivent pas être utilisées par les autres acteurs. (Un autre exemple de telles conventions est le RFC 3192 avec ses adresses de fax comme `FAX=+12027653000/T33S=1387@fax.example.org`.) Aujourd'hui, ces adresses sont utilisées bien au delà du courrier, par exemple, bien des sites Web utilisent ces adresses comme identifiants pour leurs clients enregistrés (c'est le cas d'Amazon, par exemple).

— Une autre identité est le nom de domaine, la partie à droite du @ dans une adresse. Il suit les règles générales des noms de domaine (plusieurs composants séparés par des points, etc).
— L'identificateur du message (section 3.4.1). Chaque message a un identificateur unique, stocké dans le champ `Message-ID:` et dont la forme syntaxique ressemble à une adresse de courrier (mais ne l'est pas). Il est fixé par l'origine. Un exemple est `<alpine.BSF.2.00.0907081901140.79224@in`
Cet identificateur sert à désigner un message sans ambiguité. C'est par exemple lui qui est utilisé comme référence dans les fils de discussion `<https://www.bortzmeyer.org/ne-pas-voler-les-fils`
`html>`. Une question intéressante est de savoir à partir de quand affecter un nouvel identificateur lorsque le message est modifié. Le RFC ne fournit pas de règles strictes mais suggère que, si le changement ne met en jeu que la forme (par exemple un nouvel encodage), alors, il ne s'agit pas d'un nouveau message et on ne devrait pas mettre un nouvel identificateur. Mais il y a bien d'autres cas plus douteux. Le problème est analogue à celui qui se pose pour d'autres identificateurs comme les ISBN. Avec ces derniers, on constate en pratique que les éditeurs ont des pratiques très différentes quant à la réutilisation d'un ISBN.

Notre RFC se déplace ensuite vers des entités plus concrètes, les programmes serveurs (section 4). Illustrés page 24, leur présentation nécessite d'abord un petit détour sur le format des messages (section 4.1). Il y a le message proprement dit, normalisé dans le RFC 5322. Et il y a des métadonnées, couramment appelées l'**enveloppe** (section 4.1.1) et qui servent au MHS pour assurer une distribution correcte. Les informations de l'enveloppe sont souvent enregistrées dans le message, pour analyse et déboguage "*a posteriori*" (c'est le cas des champs `Received:`).

Dans le message lui-même, il y a deux parties, les **en-têtes** (section 4.1.2) et le **corps** (section 4.1.3). Les en-têtes (comme `From:`, `Date:` ou `Subject:`) sont structurées, pour permettre un traitement automatique, par exemple par Sieve. La liste des champs est décrite dans le RFC 4021 et on peut la modifier suivant les procédures du RFC 3864. Le corps du message n'est pas normalement structuré (mais la norme MIME lui a donné un certaine structure). Les informations de l'en-tête ne coïncident pas forcément avec celles de l'enveloppe (par exemple, si un message est envoyé à une liste de diffusion, le champ `To:` de l'en-tête indique la liste, le paramètre `RCPT TO` de l'enveloppe indiquera un destinataire individuel).

Enfin, il existe aussi des « méta-messages », automatiquement générés et analysables par un programme comme les MSN ("*Message Disposition Notification*") des RFC 3297 et RFC 8098, ou comme les DSN ("*Delivery Status Notification*") du RFC 3461, qui sont typiquement utilisés comme avis de non-remise d'un message.

Pour toutes ces parties d'un message, il existe des identités différentes. Une question aussi simple que « Quel est l'auteur du message ? » a ainsi plusieurs réponses possibles, toutes légitimes. Ce point est souvent mal compris par les utilisateurs, par exemple lorsqu'une technique d'authentification est utilisée. Qu'authentifie t-elle exactement ? Pas forcément ce que l'utilisateur croit...

La section 4.1.4 fait la liste de ces identités, en indiquant qui la met dans le message. Par exemple, `RFC5322.From` est le champ `From:` de l'en-tête, et est mis par l'Auteur. `RFC5321.EHLO` est le nom qu'annonce un serveur SMTP à ses pairs et est mis par l'Origine (MSA ou MTA). `RFC2919.ListID` est l'identité d'une liste de diffusion (cf. RFC 2919) et est mise par l'Intermédiaire. Dernier exemple (mais la liste du RFC est bien plus longue) : `RFC791.SourceAddr` est l'adresse IPv4 de la machine cliente (certains protocoles, comme SPF - RFC 7208, l'utilisent).

Après cela, la section 4.2 peut commencer à lister les programmes serveurs de courrier. On y trouve le MUA ("*Mail User Agent*"), celui qui interagit directement avec l'utilisateur (Outlook, mutt, Thunderbird, etc), le MSA ("*Message Submission Agent*", le premier serveur à recevoir le courrier), le MTA ("*Mail Transfer Agent*"), ce que l'utilisateur ne voit pas, Courier, Exchange, Postfix, etc)... Pour chacun d'eux, le RFC indique les identités qui sont pertinentes pour ce serveur. Par exemple, le MSA et le MTA manipulent des identités du RFC 5321, ce qui n'est pas le cas du MUA.

Cette section couvre en détail leurs fonctions. Par exemple, le MTA a droit à une section 4.3.2 qui précise son fonctionnement de « routeur de niveau 7 ». Comme tous les routeurs, il a besoin de recevoir l'information sur les routes existantes et, sur l'Internet, cela se fait essentiellement via les enregistrements MX du DNS.

Moins connu que le MUA et le MTA, le MDA fait l'objet de la section 4.3.3. Le "*Mail Delivery Agent*" est chargé du passage du message depuis le MHS jusqu'à la boîte aux lettres de l'utilisateur. Le courrier est reçu via le protocole LMTP (RFC 2033) ou bien par un mécanisme non normalisé (par exemple, sur Unix, via un appel d'un exécutable et transmission du courrier sur l'entrée standard de celui-ci). Parmi les plus connus, dans le monde Unix, procmail ou le `local` de Postfix.

L'acheminement du courrier suit en général un modèle de "*push*" où l'emetteur envoie le message vers un destinataire supposé toujours prêt. Mais le modèle du courrier permet également un fonctionnement en "*pull*", avec des protocoles comme ceux du RFC 1985 ou du RFC 2645. (Les protocoles comme UUCP étant traités via un système de passerelle, cf. section 5.4). De même, une fois le message délivré, le destinataire peut y accéder selon un mode "*pull*", avec POP (RFC 1939) ou IMAP (RFC 3501).

Comme tout ce RFC 5598, cette section parle d'**architecture**, pas de **mise en œuvre**. Dans la pratique, une implémentation donnée de cette architecture peut répartir ses serveurs de façon assez différente (section 4.5). Par exemple, il est fréquent que les fonctions de MSA et de MTA soient fusionnées dans le même serveur (et, avec Postfix, il est même difficile de séparer les deux fonctions <`https://www.bortzmeyer.org/postfix-sasl.html`>).

La complexité des fonctions assurées par les **intermédiaires** ("*mediators*") méritait bien une section entière, la 5. Contrairement au MTA, transporteur neutre, qui doit transmettre un message avec le minimum d'altérations, l'intermédiaire a le droit de réécrire partiellement le message. Par contre, contrairement au MUA, l'intermédiaire n'est pas censé composer de messages nouveaux, et doit toujours partir d'un message existant dont il préserve le sens général. Le RFC cite plusieurs exemples d'intermédiaires :

— Les **alias** (comme gérés par les fichiers `/.forward` de Postfix et sendmail). Le destinataire décide alors de renvoyer le message à une nouvelle adresse (section 5.1). La fonction est typiquement mise en œuvre par le MDA. Le message est inchangé mais l'enveloppe indique désormais un nouveau destinataire (donc, l'identité `RFC5322.To` ne change pas mais `RFC5321.RcptTo` oui.) À noter que l'émetteur (`RFC522.From` ou bien `RFC5321.MailFrom`) n'est pas affecté donc les messages d'erreur n'iront pas au responsable de l'alias, ce qui en rend le déboguage très difficile. On peut contester ce classement des alias parmi les intermédiaires, puisque le message n'est pas modifié, mais notre RFC décide que le changement de destinataire est un changement de sémantique suffisant pour que cette fonction ne soit pas considérée comme faisant partie du MTA.

— Les listes de diffusion sont sans doute l'exemple le plus connu d'intermédiaire. Elles font l'objet de la section 5.3. Recevant un message, elles le retransmettent à plusieurs destinataires, souvent après de légères modifications (comme l'ajout d'instructions de désabonnement ou la suppression de certaines pièces jointes). Les RFC 2369 et RFC 2919 créent des identités spécifiques aux listes de diffusion, qu'on retrouve dans les en-têtes (comme `List-Id: IETF IP Performance Metrics Working Group <ippm.ietf.org>`). Beaucoup de listes de diffusion ajoutent également un champ `Reply-To:`, ce qui est en général une très mauvaise idée <`http://www.unicom.com/pw/reply-to-harmful.html`>.

— Un autre type d'intermédiaire est devenu plus rare avec le temps. À une époque, il y avait plusieurs systèmes de courrier, en général reposant sur des protocoles privés, et que certaines entreprises déployaient. C'était bien sûr ridicule (imagine t-on, sauf pour les militaires, de déployer un système de téléphonie fermé, reposant sur des protocoles spécifiques ?) mais, de la fin des années 1970 au début des années 1990, la pression idéologique était forte pour rejeter les protocoles Internet, gratuitement accessibles donc pas sérieux. Comme l'intérêt de tout système de courrier

est dans le fait qu'on peut joindre des personnes différentes, il a donc fallu développer des **passerelles** ("*gateways*"). À une époque, la seule façon d'envoyer du courrier entre les deux systèmes de messagerie privés d'IBM était via une passerelle Internet... Les passerelles sont exposées en section 5.4. Comme, par définition, elles communiquement avec un monde non-Internet, une réécriture du message, parfois avec perte d'informations, est inévitable. Tout l'art de la conception d'une passerelle est donc de minimiser ces pertes. Il est amusant de noter qu'il existe des passerelles normalisées pour des protocoles comme le fax (RFC 4143) ou la voix (RFC 3801), ce que fait aujourd'hui, par exemple, la fonction répondeur téléphonique de la Freebox).

— Un autre cas intéressant d'intermédiaire est le filtre (section 5.5), qui supprime des parties du message, par exemple parce qu'elles contiennent un virus MS-Windows (un logiciel comme Amavis `<http://www.ijs.si/software/amavisd/>` permet de créer de tels filtres).

Par rapport aux débuts du courrier électronique, un des grands changements a concerné la sécurité, un sujet devenu bien plus important aujourd'hui. D'où la section 6.1, qui lui est consacrée. Elle rappelle que le MHS n'est pas obligatoirement sécurisé. Son but est en effet de permettre l'échange de courrier avec le moins d'embêtements possibles. Sans cette propriété, le courrier électronique n'aurait jamais décollé. Mais elle a aussi été exploitée par des méchants comme les spammeurs. Cependant, il existe de nombreuses techniques pour sécuriser le courrier, par exemple TLS (RFC 3207) pour sécuriser la communication entre deux serveurs, l'authentification SMTP (RFC 4954) pour s'assurer de l'identité d'un utilisateur ou encore PGP (RFC 4880) pour protéger le message de bout en bout.

Enfin, dernier gros morceau, l'internationalisation, en section 6.2. Si les RFC de base du courrier supposent uniquement l'utilisation d'ASCII, plusieurs extensions ont permis d'avoir aujourd'hui un courrier complètement internationalisé :

— MIME (RFC 2045, RFC 2046, RFC 2047 et RFC 2049) a permis d'utiliser d'autres caractères que ceux d'ASCII dans le corps des messages et, via un surencodage, dans les en-têtes.
— Le RFC 1652 a permis de transmettre des caractères de 8 bits (extension SMTP `8BITMIME`).
— Une série de RFC `<https://www.bortzmeyer.org/courrier-entierement-internationalise.html>` a permis l'utilisation d'adresses de courrier non limitées à l'ASCII.

Merci à Olivier Miakinen pour ses nombreuses corrections.