Network Working Group
Request for Comments: 5598
Category: Informational

D. Crocker
Brandenburg InternetWorking

July 2009

Internet Mail Architecture

Abstract

Over its thirty-five-year history, Internet Mail has changed significantly in scale and complexity, as it has become a global infrastructure service. These changes have been evolutionary, rather than revolutionary, reflecting a strong desire to preserve both its installed base and its usefulness. To collaborate productively on this large and complex system, all participants need to work from a common view of it and use a common language to describe its components and the interactions among them. But the many differences in perspective currently make it difficult to know exactly what another participant means. To serve as the necessary common frame of reference, this document describes the enhanced Internet Mail architecture, reflecting the current service.

Status of This Memo

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

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

Table of Contents

1. Introduction

Over its thirty-five-year history, Internet Mail has changed significantly in scale and complexity, as it has become a global infrastructure service. These changes have been evolutionary, rather than revolutionary, reflecting a strong desire to preserve both its installed base and its usefulness. Today, Internet Mail is distinguished by many independent operators, many different components for providing service to Users, as well as many different components that transfer messages.

The underlying technical standards for Internet Mail comprise a rich array of functional capabilities. These specifications form the core:

* Simple Mail Transfer Protocol (SMTP) ([RFC0821], [RFC2821], [RFC5321]) moves a message through the Internet.

* Internet Mail Format (IMF) ([RFC0733], [RFC0822], [RFC2822], [RFC5322]) defines a message object.

* Multipurpose Internet Mail Extensions (MIME) [RFC2045] defines an enhancement to the message object that permits using multimedia attachments.

Public collaboration on technical, operations, and policy activities of email, including those that respond to the challenges of email abuse, has brought a much wider range of participants into the technical community. To collaborate productively on this large and complex system, all participants need to work from a common view of it and use a common language to describe its components and the interactions among them. But the many differences in perspective currently make it difficult to know exactly what another participant means.

It is the need to resolve these differences that motivates this document, which describes the realities of the current system. Internet Mail is the subject of ongoing technical, operations, and policy work, and the discussions often are hindered by different models of email-service design and different meanings for the same terms.

To serve as the necessary common frame of reference, this document describes the enhanced Internet Mail architecture, reflecting the current service. The document focuses on:

* Capturing refinements to the email model

* Clarifying functional roles for the architectural components

* Clarifying identity-related issues, across the email service

* Defining terminology for architectural components and their interactions

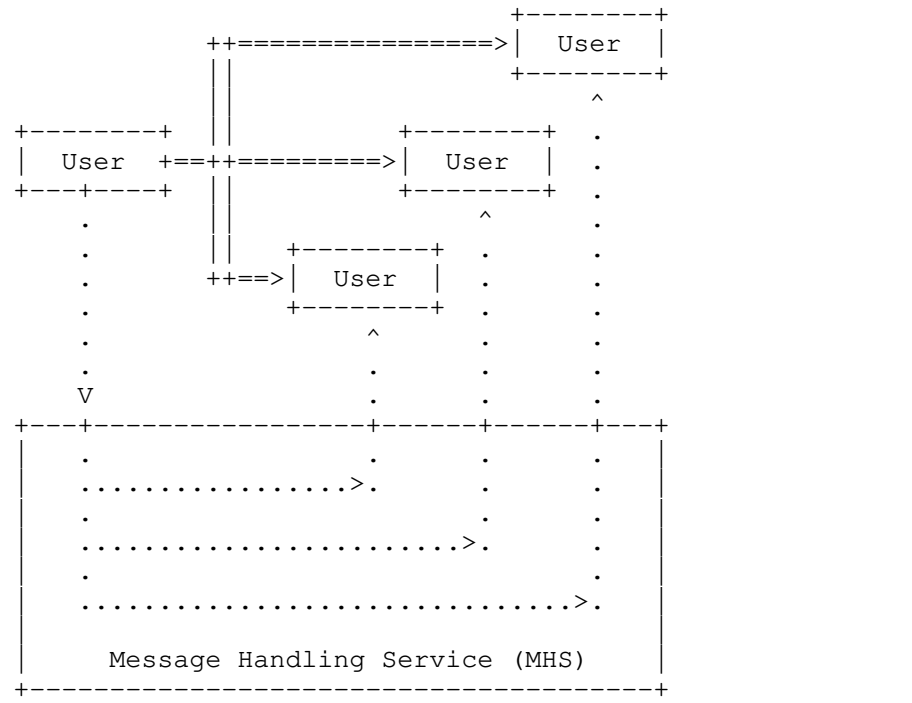
1.1. History

The first standardized architecture for networked email specified a simple split between the user world, in the form of Message User Agents (MUAs), and the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTAs) [RFC1506]. The MHS accepts a message from one User and delivers it to one or more other Users, creating a virtual MUA-to-MUA exchange environment.

As shown in Figure 1, this architecture defines two logical layers of interoperability. One is directly between Users. The other is among the components along the transfer path. In addition, there is interoperability between the layers, first when a message is posted from the User to the MHS and later when it is delivered from the MHS to the User.

The operational service has evolved, although core aspects of the service, such as mailbox addressing and message format style, remain remarkably constant. The original distinction between the user level and transfer level remains, but with elaborations in each. The term "Internet Mail" is used to refer to the entire collection of user and transfer components and services.

For Internet Mail, the term "end-to-end" usually refers to a single posting and the set of deliveries that result from a single transit of the MHS. A common exception is group dialogue that is mediated through a Mailing List; in this case, two postings occur before intended Recipients receive an Author's message, as discussed in Section 2.1.4. In fact, some uses of email consider the entire email service, including Author and Recipient, as a subordinate component. For these services, "end-to-end" refers to points outside the email service. Examples are voicemail over email [RFC3801], EDI (Electronic Data Interchange) over email [RFC1767], and facsimile over email [RFC4142].



Legend: === lines indicate primary (possibly indirect) transfers or roles
... lines indicate supporting transfers or roles

Figure 1: Basic Internet Mail Service Model

End-to-end Internet Mail exchange is accomplished by using a standardized infrastructure with these components and characteristics:

* An email object

* Global addressing

* An asynchronous sequence of point-to-point transfer mechanisms

* No requirement for prior arrangement between MTAs or between Authors and Recipients

* No requirement for prior arrangement between point-to-point transfer services over the open Internet

* No requirement for Author, Originator, or Recipients to be online at the same time

The end-to-end portion of the service is the email object, called a "message". Broadly, the message itself distinguishes control information, for handling, from the Author's content.

A precept to the design of mail over the open Internet is permitting User-to-User and MTA-to-MTA interoperability without prior, direct arrangement between the independent administrative authorities responsible for handling a message. All participants rely on having the core services universally supported and accessible, either directly or through Gateways that act as translators between Internet Mail and email environments conforming to other standards. Given the importance of spontaneity and serendipity in interpersonal communications, not requiring such prearrangement between participants is a core benefit of Internet Mail and remains a core requirement for it.

Within localized networks at the edge of the public Internet, prior administrative arrangement often is required and can include access control, routing constraints, and configuration of the information query service. Although Recipient authentication has usually been required for message access since the beginning of Internet Mail, in recent years it also has been required for message submission. In these cases, a server validates the client's identity, whether by explicit security protocols or by implicit infrastructure queries to identify "local" participants.

1.2. The Role of This Architecture

An Internet service is an integration of related capabilities among two or more participating nodes. The capabilities are accomplished across the Internet by one or more protocols. What connects a protocol to a service is an architecture. An architecture specifies how the protocols implement the service by defining the logical components of a service and the relationships among them. From that logical view, a service defines what is being done, an architecture defines where the pieces are (in relation to each other), and a protocol defines how particular capabilities are performed.

As such, an architecture will more formally describe a service at a relatively high level. A protocol that implements some portion of a service will conform to the architecture to a greater or lesser extent, depending on the pragmatic tradeoffs they make when trying to implement the architecture in the context of real-world constraints. Failure to precisely follow an architecture is not a failure of the protocol, nor is failure to precisely cast a protocol a failure of

the architecture. Where a protocol varies from the architecture, it will of course be appropriate for it to explain the reason for the variance. However, such variance is not a mark against a protocol: Happily, the IETF prefers running code to architectural purity.

In this particular case, this architecture attempts to define the logical components of Internet email and does so post hoc, trying to capture the architectural principles that the current email protocols embody. To different extents, email protocols will conform to this architecture more or less well. Insofar as this architecture differs from those protocols, the reasons are generally well understood and are required for interoperation. The differences are not a sign that protocols need to be fixed. However, this architecture is a best attempt at a logical model of Internet email, and insofar as new protocol development varies from this architecture, it is necessary for designers to understand those differences and explain them carefully.

1.3. Document Conventions

References to structured fields of a message use a two-part dotted notation. The first part cites the document that contains the specification for the field, and the second part is the name of the field. Hence <RFC5322.From> is the IMF From: header field in an email content header, and <RFC5321.MailFrom> is the address in the SMTP "Mail From" command.

When occurring without the IMF (RFC 5322) qualifier, header field names are shown with a colon suffix. For example, From:.

References to labels for actors, functions or components have the first letter capitalized.

2. Responsible Actor Roles

Internet Mail is a highly distributed service, with a variety of Actors playing different roles. These Actors fall into three basic types:

* User

* Message Handling Service (MHS)

* ADministrative Management Domain (ADMD)

Although related to a technical architecture, the focus on Actors concerns participant responsibilities, rather than functionality of modules. For that reason, the labels used are different from those used in classic diagrams of email architecture.

2.1. User Actors

Users are the sources and sinks of messages. Users can be people, organizations, or processes. They can have an exchange that iterates, and they can expand or contract the set of Users that participate in a set of exchanges. In Internet Mail, there are four types of Users:

* Authors
* Recipients
* Return Handlers
* Mediators

Figure 2 shows the primary and secondary flows of messages among them. As a pragmatic heuristic: User Actors can generate, modify, or look at the whole message.

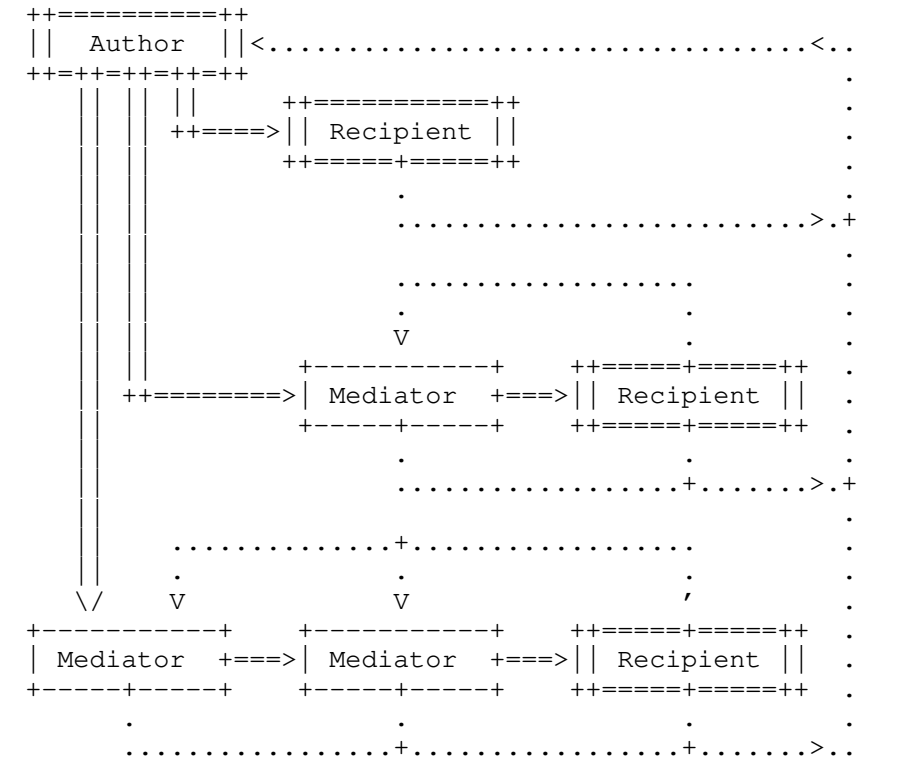


Figure 2: Relationships among User Actors

From a User's perspective, all message-transfer activities are performed by a monolithic Message Handling Service (MHS), even though the actual service can be provided by many independent organizations. Users are customers of this unified service.

Whenever any MHS Actor sends information back to an Author or Originator in the sequence of handling a message, that Actor is a User.

2.1.1. Author

The Author is responsible for creating the message, its contents, and its list of Recipient addresses. The MHS transfers the message from the Author and delivers it to the Recipients. The MHS has an Originator role (Section 2.2.1) that correlates with the Author role.

2.1.2. Recipient

The Recipient is a consumer of the delivered message. The MHS has a Receiver role (Section 2.2.4) that correlates with the Recipient role. This is labeled Recv in Figure 3.

Any Recipient can close the user-communication loop by creating and submitting a new message that replies to the Author. An example of an automated form of reply is the Message Disposition Notification (MDN), which informs the Author about the Recipient's handling of the message. (See Section 4.1.)

2.1.3. Return Handler

Also called "Bounce Handler", the Return Handler is a special form of Recipient tasked with servicing notifications generated by the MHS as it transfers or delivers the message. (See Figure 3.) These notices can be about failures or completions and are sent to an address that is specified by the Originator. This Return Handling address (also known as a Return Address) might have no visible characteristics in common with the address of the Author or Originator.

2.1.4. Mediator

A Mediator receives, aggregates, reformulates, and redistributes messages among Authors and Recipients who are the principals in (potentially) protracted exchanges. This activity is easily confused with the underlying MHS transfer exchanges. However, each serves very different purposes and operates in very different ways.

When mail is delivered to the Mediator specified in the RFC5321.RcptTo command for the original message, the MHS handles it the same way as for any other Recipient. In particular, the MHS sees each posting and delivery activity between sources and sinks as independent; it does not see subsequent re-posting as a continuation of a process. Because the Mediator originates messages, it can receive replies. Hence, when submitting a reformulated message, the Mediator is an Author, albeit an Author actually serving as an agent of one or more other Authors. So a Mediator really is a full-fledged User. Mediators are considered extensively in Section 5.

A Mediator attempts to preserve the original Author's information in the message it reformulates but is permitted to make meaningful changes to the message content or envelope. The MHS sees a new message, but Users receive a message that they interpret as being from, or at least initiated by, the Author of the original message. The role of a Mediator is not limited to merely connecting other participants; the Mediator is responsible for the new message.

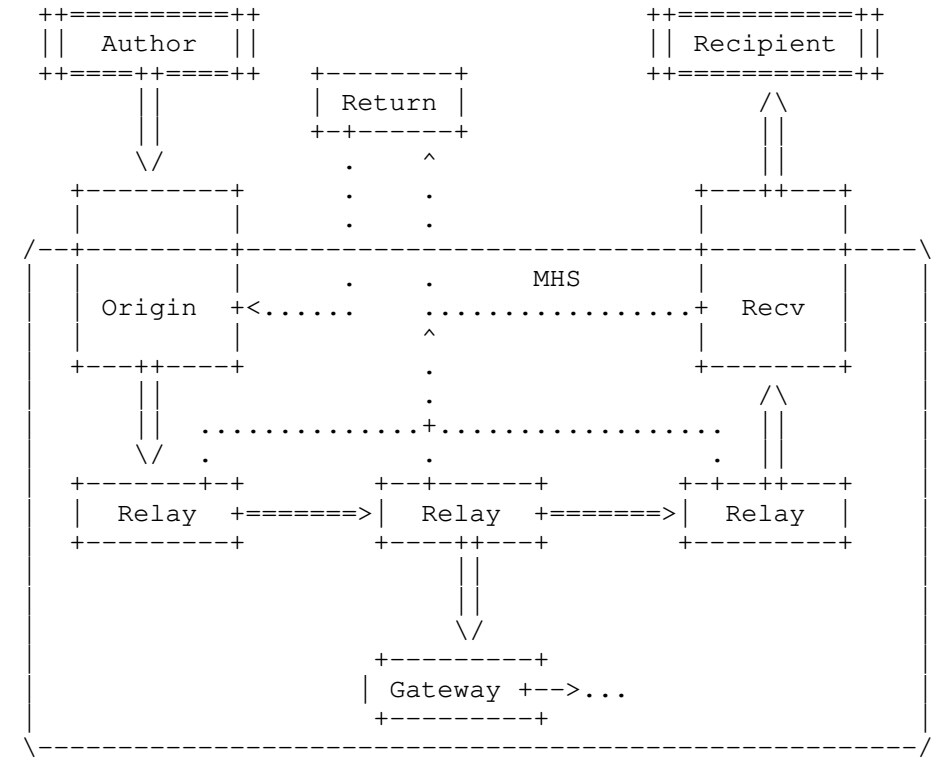
A Mediator's role is complex and contingent, for example, modifying and adding content or regulating which Users are allowed to participate and when. The common example of this role is a group Mailing List. In a more complex use, a sequence of Mediators could perform a sequence of formal steps, such as reviewing, modifying, and approving a purchase request.

A Gateway is a particularly interesting form of Mediator. It is a hybrid of User and Relay that connects heterogeneous mail services. Its purpose is to emulate a Relay. For a detailed discussion, see Section 2.2.3.

2.2. Message Handling Service (MHS) Actors

The Message Handling Service (MHS) performs a single end-to-end transfer on behalf of the Author to reach the Recipient addresses specified in the original RFC5321.RcptTo commands. Exchanges that are either mediated or iterative and protracted, such as those used for collaboration over time, are handled by the User Actors, not by the MHS Actors. As a pragmatic heuristic MHS Actors generate, modify, or look at only transfer data, rather than the entire message.

Figure 3 shows the relationships among transfer participants in Internet Mail. Although it shows the Originator (labeled Origin) as distinct from the Author, and Receiver (labeled Recv) as distinct from Recipient, each pair of roles usually has the same Actor. Transfers typically entail one or more Relays. However, direct delivery from the Originator to Receiver is possible. Intra-organization mail services usually have only one Relay.



Legend: === and || lines indicate primary (possibly indirect) transfers or roles
... lines indicate supporting transfers or roles

Figure 3: Relationships among MHS Actors

2.2.1. Originator

The Originator ensures that a message is valid for posting and then submits it to a Relay. A message is valid if it conforms to both Internet Mail standards and local operational policies. The Originator can simply review the message for conformance and reject it if it finds errors, or it can create some or all of the necessary information. In effect, the Originator is responsible for the functions of the Mail Submission Agent.

The Originator operates with dual allegiance. It serves the Author and can be the same entity. But its role in assuring validity means that it also represents the local operator of the MHS, that is, the local ADministrative Management Domain (ADMD).

The Originator also performs any post-submission, Author-related administrative tasks associated with message transfer and delivery. Notably, these tasks pertain to sending error and delivery notices, enforcing local policies, and dealing with messages from the Author that prove to be problematic for the Internet. The Originator is accountable for the message content, even when it is not responsible for it. The Author creates the message, but the Originator handles any transmission issues with it.

2.2.2. Relay

The Relay performs MHS-level transfer-service routing and store-and-forward by transmitting or retransmitting the message to its Recipients. The Relay adds trace information [RFC2505] but does not modify the envelope information or the message content semantics. It can modify message content representation, such as changing the form of transfer encoding from binary to text, but only as required to meet the capabilities of the next hop in the MHS.

A Message Handling System (MHS) network consists of a set of Relays. This network is above any underlying packet-switching network that might be used and below any Gateways or other Mediators.

In other words, email scenarios can involve three distinct architectural layers, each providing its own type of data of store-and-forward service:

* User Mediators

* MHS Relays

* Packet Switches

The bottom layer is the Internet's IP service. The most basic email scenarios involve Relays and Switches.

When a Relay stops attempting to transfer a message, it becomes an Author because it sends an error message to the Return Address. The potential for looping is avoided by omitting a Return Address from this message.

2.2.3. Gateway

A Gateway is a hybrid of User and Relay that connects heterogeneous mail services. Its purpose is to emulate a Relay and the closer it comes to this, the better. A Gateway operates as a User when it needs the ability to modify message content.

Differences between mail services can be as small as minor syntax variations, but they usually encompass significant, semantic distinctions. One difference could be email addresses that are hierarchical and machine-specific rather than a flat, global namespace. Another difference could be support for text-only content or multimedia. Hence the Relay function in a Gateway presents a significant design challenge if the resulting performance is to be seen as nearly seamless. The challenge is to ensure User-to-User functionality between the services, despite differences in their syntax and semantics.

The basic test of Gateway design is whether an Author on one side of a Gateway can send a useful message to a Recipient on the other side, without requiring changes to any components in the Author's or Recipient's mail services other than adding the Gateway. To each of these otherwise independent services, the Gateway appears to be a native participant. But the ultimate test of Gateway design is whether the Author and Recipient can sustain a dialogue. In particular, can a Recipient's MUA automatically formulate a valid Reply that will reach the Author?

2.2.4. Receiver

The Receiver performs final delivery or sends the message to an alternate address. It can also perform filtering and other policy enforcement immediately before or after delivery.

2.3. Administrative Actors

Administrative Actors can be associated with different organizations, each with its own administrative authority. This operational independence, coupled with the need for interaction between groups, provides the motivation to distinguish among ADministrative Management Domains (ADMDs). Each ADMD can have vastly different operating policies and trust-based decision-making. One obvious example is the distinction between mail that is exchanged within an organization and mail that is exchanged between independent organizations. The rules for handling both types of traffic tend to be quite different. That difference requires defining the boundaries of each, and this requires the ADMD construct.

Operation of Internet Mail services is carried out by different providers (or operators). Each can be an independent ADMD. This independence of administrative decision-making defines boundaries that distinguish different portions of the Internet Mail service. A department that operates a local Relay, an IT department that operates an enterprise Relay, and an ISP that operates a public shared email service can be configured into many combinations of

administrative and operational relationships. Each is a distinct ADMD, potentially having a complex arrangement of functional components. Figure 4 depicts relationships among ADMDs. The benefit of the ADMD construct is that it facilitates discussion about designs, policies, and operations that need to distinguish between internal issues and external ones.

The architectural impact of the need for boundaries between ADMDs is discussed in [Tussle]. Most significant is that the entities communicating across ADMD boundaries typically have the added burden of enforcing organizational policies concerning external communications. At a more mundane level, routing mail between ADMDs can be an issue, such as needing to route mail between organizational partners over specially trusted paths.

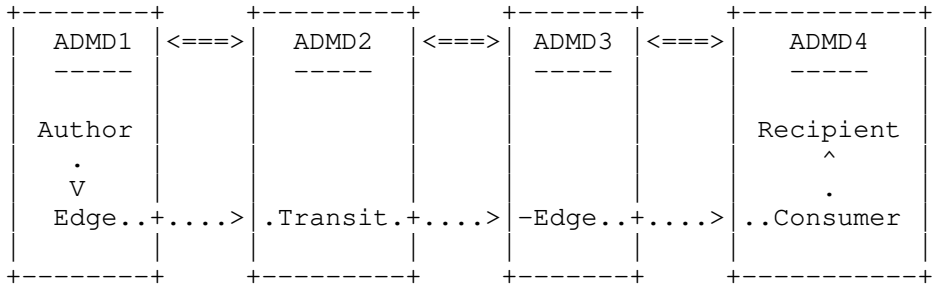
These are three basic types of ADMDs:

Edge: Independent transfer services in networks at the edge of the open Internet Mail service.

Consumer: Might be a type of Edge service, as is common for web-based email access.

Transit: Mail Service Providers (MSPs) that offer value-added capabilities for Edge ADMDs, such as aggregation and filtering.

The mail-level transit service is different from packet-level switching. End-to-end packet transfers usually go through intermediate routers; email exchange across the open Internet can be directly between the Boundary MTAs of Edge ADMDs. This distinction between direct and indirect interaction highlights the differences discussed in Section 2.2.2.



Legend: === lines indicate primary (possibly indirect) transfers or roles
... lines indicate supporting transfers or roles

Figure 4: Administrative Domain (ADMD) Example

Edge networks can use proprietary email standards internally. However, the distinction between Transit network and Edge network transfer services is significant because it highlights the need for concern over interaction and protection between independent administrations. In particular, this distinction calls for additional care in assessing the transitions of responsibility and the accountability and authorization relationships among participants in message transfer.

The interactions of ADMD components are subject to the policies of that domain, which cover concerns such as these:

* Reliability
* Access control
* Accountability
* Content evaluation and modification

These policies can be implemented in different functional components, according to the needs of the ADMD. For example, see [RFC5068].

Consumer, Edge, and Transit services can be offered by providers that operate component services or sets of services. Further, it is possible for one ADMD to host services for other ADMDs.

These are common examples of ADMDs:

Enterprise Service Providers:

These ADMDs operate the internal data and/or the mail services within an organization.

Internet Service Providers (ISP):

These ADMDs operate the underlying data communication services, which are used by one or more Relay and User. ISPs are not responsible for performing email functions, but they can provide an environment in which those functions can be performed.

Mail Service Providers:

These ADMDs operate email services, such as for consumers or client companies.

Practical operational concerns demand that providers be involved in administration and enforcement issues. This involvement can extend to operators of lower-level packet services.

3. Identities

The forms of identity used by Internet Mail are: mailbox, domain name, message-ID, and ENVID (envelope identifier). Each is globally unique.

3.1. Mailbox

> "A mailbox receives mail. It is a conceptual entity that does not necessarily pertain to file storage." [RFC5322]

A mailbox is specified as an Internet Mail address <addr-spec>. It has two distinct parts, separated by an at-sign (@). The right side is a globally interpreted domain name associated with an ADMD. Domain names are discussed in Section 3.3. Formal Internet Mail addressing syntax can support source routes to indicate the path through which a message ought to be sent. The use of source routes is not common and has been deprecated in [RFC5321].

The portion to the left of the at-sign contains a string that is globally opaque and is called the <local-part>. It is interpreted only by the entity specified by the address's domain name. Except as noted later in this section, all other entities treat the <local-part> as an uninterpreted literal string and preserve all

of its original details. As such, its public distribution is equivalent to sending a Web browser "cookie" that is only interpreted upon being returned to its creator.

Some local-part values have been standardized for contacting personnel at an organization. These names cover common operations and business functions [RFC2142].

It is common for sites to have local structuring conventions for the left-hand side, <local-part>, of an <addr-spec>. This permits sub-addressing, such as for distinguishing different discussion groups used by the same participant. However, it is worth stressing that these conventions are strictly private to the User's organization and are not interpreted by any domain except the one listed in the right side of the <addr-spec>. The exceptions are those specialized services that conform to public, standardized conventions, as noted below.

Basic email addressing defines the <local-part> as being globally opaque. However, there are some uses of email that add a standardized, global schema to the value, such as between an Author and a Gateway. The <local-part> details remain invisible to the public email transfer infrastructure, but provide addressing and handling instructions for further processing by the Gateway. Standardized examples of these conventions are the telephone numbering formats for the Voice Profile for Internet Mail (VPIM) [RFC3801], such as:

+16137637582@vpim.example.com,

and iFax ([RFC3192], [RFC4143] such as:

FAX=+12027653000/T33S=1387@ifax.example.com.

3.2. Scope of Email Address Use

Email addresses are being used far beyond their original role in email transfer and delivery. In practical terms, an email address string has become the common identifier for representing online identity. Hence, it is essential to be clear about both the nature and role of an identity string in a particular context and the entity responsible for setting that string. For example, see Sections 4.1.4, 4.3.3, and 5.

3.3. Domain Names

A domain name is a global reference to an Internet resource, such as a host, a service, or a network. A domain name usually maps to one or more IP Addresses. Conceptually, the name can encompass an organization, a collection of machines integrated into a homogeneous service, or a single machine. A domain name can be administered to refer to an individual User, but this is not common practice. The name is structured as a hierarchical sequence of labels, separated by dots (.), with the top of the hierarchy being on the right end of the sequence. There can be many names in the sequence -- that is, the depth of the hierarchy can be substantial. Domain names are defined and operated through the Domain Name System (DNS) ([RFC1034], [RFC1035], [RFC2181]).

When not part of a mailbox address, a domain name is used in Internet Mail to refer to the ADMD or to the host that took action upon the message, such as providing the administrative scope for a message identifier or performing transfer processing.

3.4. Message Identifier

There are two standardized tags for identifying messages: Message-ID: and ENVID. A Message-ID: pertains to content, and an ENVID pertains to transfer.

3.4.1. Message-ID

IMF provides for, at most, a single Message-ID:. The Message-ID: for a single message, which is a user-level IMF tag, has a variety of uses including threading, aiding identification of duplicates, and DSN (Delivery Status Notification) tracking. The Originator assigns the Message-ID:. The Recipient's ADMD is the intended consumer of the Message-ID:, although any Actor along the transfer path can use it.

Message-ID: is globally unique. Its format is similar to that of a mailbox, with two distinct parts separated by an at-sign (@). Typically, the right side specifies the ADMD or host that assigns the identifier, and the left side contains a string that is globally opaque and serves to uniquely identify the message within the domain referenced on the right side. The duration of uniqueness for the message identifier is undefined.

When a message is revised in any way, the decision whether to assign a new Message-ID: requires a subjective assessment to determine whether the editorial content has been changed enough to constitute a new message. [RFC5322] states that "a message identifier pertains to

exactly one version of a particular message; subsequent revisions to the message each receive new message identifiers." Yet experience suggests that some flexibility is needed. An impossible test is whether the Recipient will consider the new message to be equivalent to the old one. For most components of Internet Mail, there is no way to predict a specific Recipient's preferences on this matter. Both creating and failing to create a new Message-ID: have their downsides.

Here are some guidelines and examples:

- If a message is changed only in form, such as character encoding, it is still the same message.
- If a message has minor additions to the content, such as a Mailing List tag at the beginning of the RFC5322.Subject header field, or some Mailing List administrative information added to the end of the primary body part text, it is probably the same message.
- If a message has viruses deleted from it, it is probably the same message.
- If a message has offensive words deleted from it, some Recipients will consider it the same message, but some will not.
- If a message is translated into a different language, some Recipients will consider it the same message, but some will not.
- If a message is included in a digest of messages, the digest constitutes a new message.
- If a message is forwarded by a Recipient, what is forwarded is a new message.
- If a message is "redirected", such as using IMF "Resent-*" header fields, some Recipients will consider it the same message, but some will not.

The absence of both objective, precise criteria for regenerating a Message-ID: and strong protection associated with the string means that the presence of an ID can permit an assessment that is marginally better than a heuristic, but the ID certainly has no value on its own for strict formal reference or comparison. For that reason, the Message-ID: is not intended to be used for any function that has security implications.

3.4.2. ENVID

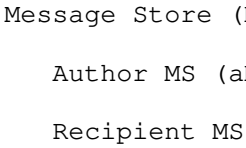
The ENVID (envelope identifier) can be used for message-tracking purposes ([RFC3885], [RFC3464]) concerning a single posting/delivery transfer. The ENVID labels a single transit of the MHS by a specific message. So, the ENVID is used for one message posting until that message is delivered. A re-posting of the message, such as by a Mediator, does not reuse that ENVID, but can use a new one, even though the message might legitimately retain its original Message-ID:.

The format of an ENVID is free form. Although its creator might choose to impose structure on the string, none is imposed by Internet standards. By implication, the scope of the string is defined by the domain name of the Return Address.

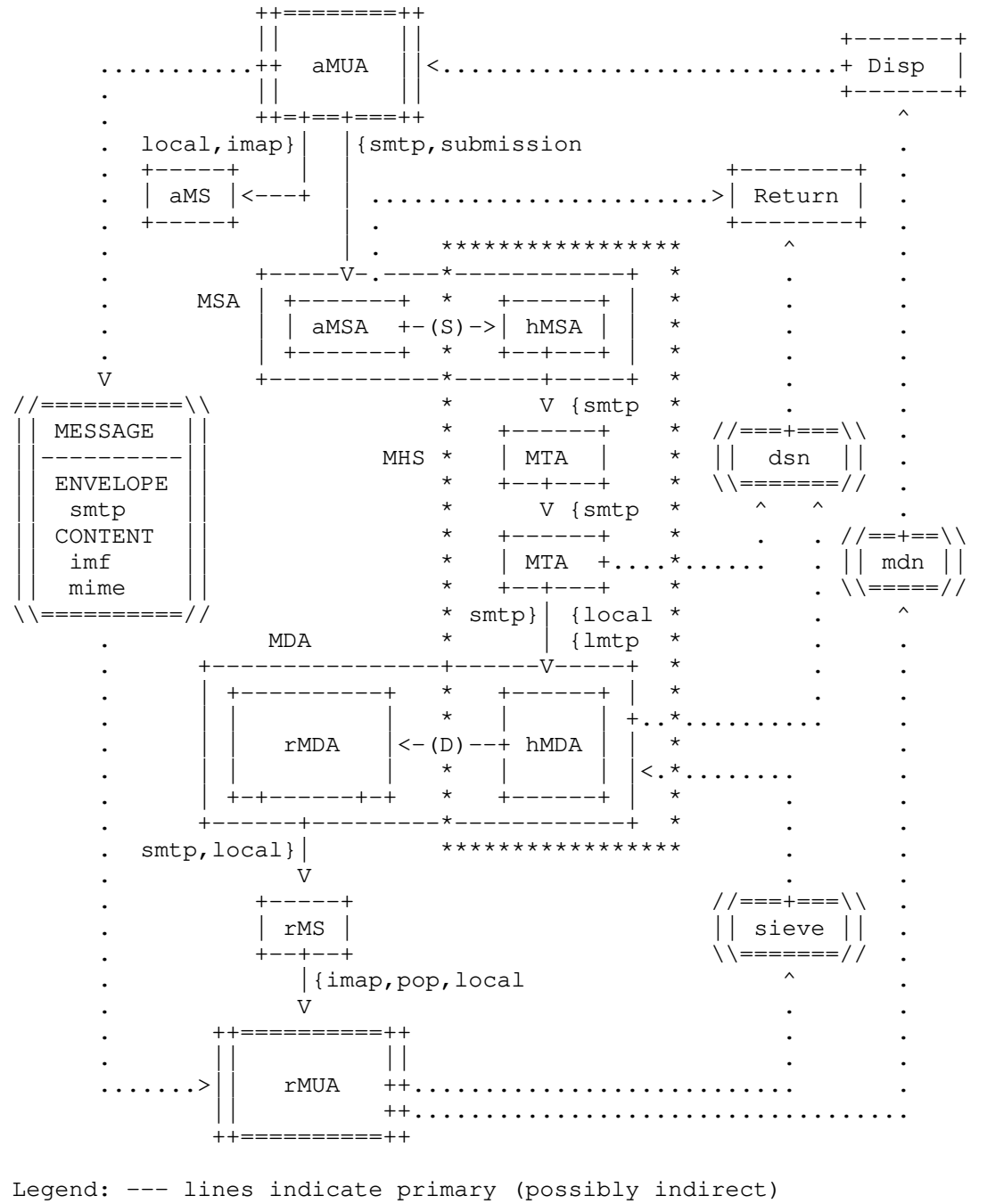
4. Services and Standards

The Internet Mail architecture comprises six basic types of functionality, which are arranged to support a store-and-forward service. As shown in Figure 5, each type can have multiple instances, some of which represent specialized roles. This section considers the activities and relationships among these components, and the Internet Mail standards that apply to them.

- Message
- Message User Agent (MUA)
  - Author MUA (aMUA)
  - Recipient MUA (rMUA)
- Message Submission Agent (MSA)
  - Author-focused MSA functions (aMSA)
  - MHS-focused MSA functions (hMSA)
- Message Transfer Agent (MTA)
- Message Delivery Agent (MDA)
  - Recipient-focused MDA functions (rMDA)
  - MHS-focused MDA functions (hMDA)



This figure shows function modules and the standardized protocols used between them.



Legend: --- lines indicate primary (possibly indirect)
transfers or roles
=== boxes indicate data objects

... lines indicate supporting transfers or roles
*** lines indicate aggregated service

Figure 5: Protocols and Services

4.1. Message Data

The purpose of the Message Handling System (MHS) is to exchange an IMF message object among participants [RFC5322]. All of its underlying mechanisms serve to deliver that message from its Author to its Recipients. A message can be explicitly labeled as to its nature [RFC3458].

A message comprises a transit-handling envelope and the message content. The envelope contains information used by the MHS. The content is divided into a structured header and the body. The header comprises transit-handling trace information and structured fields that are part of the Author's message content. The body can be unstructured lines of text or a tree of multimedia subordinate objects, called "body-parts" or, popularly, "attachments". [RFC2045], [RFC2046], [RFC2047], [RFC4288], [RFC4289], [RFC2049].

In addition, Internet Mail has a few conventions for special control data, notably:

Delivery Status Notification (DSN):

A Delivery Status Notification (DSN) is a message that can be generated by the MHS (MSA, MTA, or MDA) and sent to the RFC5321.MailFrom address. MDA and MTA are shown as sources of DSNs in Figure 5, and the destination is shown as Returns. DSNs provide information about message transit, such as transfer errors or successful delivery [RFC3461].

Message Disposition Notification (MDN):

A Message Disposition Notification (MDN) is a message that provides information about post-delivery processing, such as indicating that the message has been displayed [RFC3798] or the form of content that can be supported [RFC3297]. It can be generated by an rMUA and is sent to the Disposition-Notification-To addresses. The mailbox for this is shown as Disp in Figure 5.

Message Filtering (SIEVE):

Sieve is a scripting language used to specify conditions for differential handling of mail, typically at the time of delivery [RFC5228]. Scripts can be conveyed in a variety of ways, such as a MIME part in a message. Figure 5 shows a Sieve script going from the rMUA to the MDA. However, filtering can be done at many different points along the transit path, and any one or more of them might be subject to Sieve directives, especially within a single ADMD. Figure 5 shows only one relationship, for (relative) simplicity.

4.1.1. Envelope

Internet Mail has a fragmented framework for transit-related handling information. Information that is used directly by the MHS is called the "envelope". It directs handling activities by the transfer service and is carried in transfer-service commands. That is, the envelope exists in the transfer protocol SMTP [RFC5321].

Trace information, such as RFC5322.Received, is recorded in the message header and is not subsequently altered [RFC5322].

4.1.2. Header Fields

Header fields are attribute name/value pairs that cover an extensible range of email-service parameters, structured user content, and user transaction meta-information. The core set of header fields is defined in [RFC5322]. It is common practice to extend this set for different applications. Procedures for registering header fields are defined in [RFC3864]. An extensive set of existing header field registrations is provided in [RFC4021].

One danger of placing additional information in header fields is that Gateways often alter or delete them.

4.1.3. Body

The body of a message might be lines of ASCII text or a hierarchically structured composition of multimedia body part attachments using MIME ([RFC2045], [RFC2046], [RFC2047], [RFC4288], and [RFC2049]).

4.1.4. Identity References in a Message

Table 1 lists the core identifiers present in a message during transit.

Layer	Field	Set By
Message Body	MIME Header	Author
Message header fields	From:	Author
	Sender:	Originator
	Reply-To:	Author
	To:, CC:, BCC:	Author
	Message-ID:	Originator
	Received:	Originator, Relay, Receiver
	Return-Path:	MDA, from MailFrom
	Resent-*:	Mediator
	List-Id:	Mediator
	List-*:	Mediator
SMTP	HELO/EHLO	Latest Relay Client
	ENVID	Originator
	MailFrom	Originator
	RcptTo	Author
	ORCPT	Originator
IP	Source Address	Latest Relay Client

Legend:

Layer - The part of the email architecture that uses the identifier.

Field - The protocol construct that contains the identifier.

Set By - The Actor role responsible for specifying the identifier value (and this can be different from the Actor that performs the fill-in function for the protocol construct).

Table 1: Layered Identities

These are the most common address-related fields:

RFC5322.From: Set by - Author

Names and addresses for Authors of the message content are listed in the From: field.

RFC5322.Reply-To: Set by - Author

If a Recipient sends a reply message that would otherwise use the RFC5322.From field addresses in the original message, the addresses in the RFC5322.Reply-To field are used instead. In other words, this field overrides the From: field for responses from Recipients.

RFC5322.Sender: Set by - Originator

This field specifies the address responsible for submitting the message to the transfer service. This field can be omitted if it contains the same address as RFC5322.From. However, omitting this field does not mean that no Sender is specified; it means that that header field is virtual and that the address in the From: field is to be used.

Specification of the notifications Return Addresses, which are contained in RFC5321.MailFrom, is made by the RFC5322.Sender. Typically, the Return address is the same as the Sender address. However, some usage scenarios require it to be different.

RFC5322.To/.CC: Set by - Author

These fields specify MUA Recipient addresses. However, some or all of the addresses in these fields might not be present in the RFC5321.RcptTo commands.

The distinction between To and CC is subjective. Generally, a To addressee is considered primary and is expected to take action on the message. A CC addressee typically receives a copy as a courtesy.

RFC5322.BCC: Set by - Author

A copy of the message might be sent to an addressee whose participation is not to be disclosed to the RFC5322.To or RFC5322.CC Recipients and, usually, not to the other BCC Recipients. The BCC: header field indicates a message copy to such a Recipient. Use of this field is discussed in [RFC5322].

RFC5321.HELO/.EHLO: Set by - Originator, MSA, MTA

Any SMTP client -- including Originator, MSA, or MTA -- can specify its hosting domain identity for the SMTP HELO or EHLO command operation.

RFC3461.ENVID: Set by - Originator

The MSA can specify an opaque string, to be included in a DSN, as a means of assisting the Return Address Recipient in identifying the message that produced a DSN or message tracking.

RFC5321.MailFrom: Set by - Originator

This field is an end-to-end string that specifies an email address for receiving return control information, such as returned messages. The name of this field is misleading, because it is not required to specify either the Author or the Actor responsible for submitting the message. Rather, the Actor responsible for submission specifies the RFC5321.MailFrom address. Ultimately, the simple basis for deciding which address needs to be in the RFC5321.MailFrom field is to determine which address is to be informed about transfer-level problems (and possibly successes).

RFC5321.RcptTo: Set by - Author, Final MTA, MDA

This field specifies the MUA mailbox address of a Recipient. The string might not be visible in the message content header. For example, the IMF destination address header fields, such as RFC5322.To, might specify a Mailing List mailbox, while the RFC5321.RcptTo address specifies a member of that list.

RFC5321.ORCPT: Set by - Originator.

This is an optional parameter to the RCPT command, indicating the original address to which the current RCPT TO address corresponds, after a mapping was performed during transit. An ORCPT is the only reliable way to correlate a DSN from a multi-Recipient message transfer with the intended Recipient.

RFC5321.Received: Set by - Originator, Relay, Mediator, Dest

This field contains trace information, including originating host, Relays, Mediators, and MSA host domain names and/or IP Addresses.

RFC5321.Return-Path: Set by - Originator

The MDA records the RFC5321.MailFrom address into the RFC5321.Return-Path field.

RFC2919.List-Id: Set by - Mediator, Author

This field provides a globally unique Mailing List naming framework that is independent of particular hosts [RFC2919].

The identifier is in the form of a domain name; however, the string usually is constructed by combining the two parts of an email address. The result is rarely a true domain name, listed in the domain name service, although it can be.

RFC2369.List-*: Set by - Mediator, Author

[RFC2369] defines a collection of message header fields for use by Mailing Lists. In effect, they supply list-specific parameters for common Mailing-List user operations. The identifiers for these operations are for the list itself and the user-as-subscriber [RFC2369].

RFC0791.SourceAddr: Set by - The Client SMTP sending host immediately preceding the current receiving SMTP server

[RFC0791] defines the basic unit of data transfer for the Internet: the IP datagram. It contains a Source Address field that specifies the IP Address for the host (interface) from which the datagram was sent. This information is set and provided by the IP layer, which makes it independent of mail-level mechanisms. As such, it is often taken to be authoritative, although it is possible to provide false addresses.

4.2. User-Level Services

Interactions at the user level entail protocol exchanges, distinct from those that occur at lower layers of the Internet Mail MHS architecture that is, in turn, above the Internet Transport layer. Because the motivation for email, and much of its use, is for interaction among people, the nature and details of these protocol exchanges often are determined by the needs of interpersonal and group communication. To accommodate the idiosyncratic behavior inherent in such communication, only subjective guidelines, rather than strict rules, can be offered for some aspects of system behavior. Mailing Lists provide particularly salient examples.

4.2.1. Message User Agent (MUA)

A Message User Agent (MUA) works on behalf of User Actors and User applications. It is their representative within the email service.

The Author MUA (aMUA) creates a message and performs initial submission into the transfer infrastructure via a Mail Submission Agent (MSA). It can also perform any creation- and posting-time archiving in its Message Store (aMS). An MUA aMS can organize messages in many different ways. A common model uses aggregations, called "folders"; in IMAP they are called "mailboxes". This model

allows a folder for messages under development (Drafts), a folder for messages waiting to be sent (Queued or Unsent), and a folder for messages that have been successfully posted for transfer (Sent). But none of these folders is required. For example, IMAP allows drafts to be stored in any folder, so no Drafts folder needs to be present.

The Recipient MUA (rMUA) works on behalf of the Recipient to process received mail. This processing includes generating user-level disposition control messages, displaying and disposing of the received message, and closing or expanding the user-communication loop by initiating replies and forwarding new messages.

NOTE: Although not shown in Figure 5, an MUA itself can have a distributed implementation, such as a "thin" user-interface module on a constrained device such as a smartphone, with most of the MUA functionality running remotely on a more capable server. An example of such an architecture might use IMAP [RFC3501] for most of the interactions between an MUA client and an MUA server. An approach for such scenarios is defined by [RFC4550].

A Mediator is a special class of MUA. It performs message re-posting, as discussed in Section 2.1.

An MUA can be automated, on behalf of a User who is not present at the time the MUA is active. One example is a bulk sending service that has a timed-initiation feature. These services are not to be confused with a Mailing List Mediator, since there is no incoming message triggering the activity of the automated service.

A popular and problematic MUA is an automatic responder, such as one that sends out-of-office notices. This behavior might be confused with that of a Mediator, but this MUA is generating a new message. Automatic responders can annoy Users of Mailing Lists unless they follow [RFC3834].

The identity fields are relevant to a typical MUA:

RFC5322.From

RFC5322.Reply-To

RFC5322.Sender

RFC5322.To, RFC5322.CC

RFC5322.BCC

4.2.2. Message Store (MS)

An MUA can employ a long-term Message Store (MS). Figure 5 depicts an Author's MS (aMS) and a Recipient's MS (rMS). An MS can be located on a remote server or on the same machine as the MUA.

An MS acquires messages from an MDA either proactively by a local mechanism or even by a standardized mechanism such as SMTP(!), or reactively by using POP or IMAP. The MUA accesses the MS either by a local mechanism or by using POP or IMAP. Using POP for individual message accesses, rather than for bulk transfer, is relatively rare and inefficient.

4.3. MHS-Level Services

4.3.1. Mail Submission Agent (MSA)

A Mail Submission Agent (MSA) accepts the message submitted by the aMUA and enforces the policies of the hosting ADMD and the requirements of Internet standards. An MSA represents an unusual functional dichotomy. It represents the interests of the Author (aMUA) during message posting, to facilitate posting success; it also represents the interests of the MHS. In the architecture, these responsibilities are modeled, as shown in Figure 5, by dividing the MSA into two sub-components, aMSA and hMSA, respectively. Transfer of responsibility for a single message, from an Author's environment to the MHS, is called "posting". In Figure 5, it is marked as the (S) transition, within the MSA.

The hMSA takes transit responsibility for a message that conforms to the relevant Internet standards and to local site policies. It rejects messages that are not in conformance. The MSA performs final message preparation for submission and effects the transfer of responsibility to the MHS, via the hMSA. The amount of preparation depends upon the local implementations. Examples of aMSA tasks include adding header fields, such as Date: and Message-ID:, and modifying portions of the message from local notations to Internet standards, such as expanding an address to its formal IMF representation.

Historically, standards-based MUA/MSA message postings have used SMTP [RFC5321]. The standard currently preferred is SUBMISSION [RFC4409]. Although SUBMISSION derives from SMTP, it uses a separate TCP port and imposes distinct requirements, such as access authorization.

These identities are relevant to the MSA:

RFC5321.HELO/.EHLO

RFC3461.ENVID

RFC5321.MailFrom

RFC5321.RcptTo

RFC5321.Received

RFC0791.SourceAddr

4.3.2. Message Transfer Agent (MTA)

A Message Transfer Agent (MTA) relays mail for one application-level "hop". It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the Recipients. Of course, email objects are typically much larger than the payload of a packet or datagram, and the end-to-end latencies are typically much higher. Relaying is performed by a sequence of MTAs until the message reaches a destination MDA. Hence, an MTA implements both client and server MTA functionality; it does not change addresses in the envelope or reformulate the editorial content. A change in data form, such as to MIME Content-Transfer-Encoding, is within the purview of an MTA, but removal or replacement of body content is not. An MTA also adds trace information [RFC2505].

NOTE: Within a destination ADMD, email-relaying modules can make a variety of changes to the message, prior to delivery. In such cases, these modules are acting as Gateways, rather than MTAs.

Internet Mail uses SMTP ([RFC5321], [RFC2821], [RFC0821]) primarily to effect point-to-point transfers between peer MTAs. Other transfer mechanisms include Batch SMTP [RFC2442] and On-Demand Mail Relay (ODMR) SMTP [RFC2645]. As with most network-layer mechanisms, the Internet Mail SMTP supports a basic level of reliability, by virtue of providing for retransmission after a temporary transfer failure. Unlike typical packet switches (and Instant Messaging services), Internet Mail MTAs are expected to store messages in a manner that allows recovery across service interruptions, such as host-system shutdown. The degree of such robustness and persistence by an MTA can vary. The base SMTP specification provides a framework for protocol response codes. An extensible enhancement to this framework is defined in [RFC5248].

Although quite basic, the dominant routing mechanism for Internet Mail is the DNS MX record [RFC1035], which specifies an MTA through which the queried domain can be reached. This mechanism presumes a public, or at least a common, backbone that permits any attached MTA to connect to any other.

MTAs can perform any of these well-established roles:

Boundary MTA: An MTA that is part of an ADMD and interacts with MTAs in other ADMDs. This is also called a Border MTA. There can be different Boundary MTAs, according to the direction of mail-flow.

Outbound MTA: An MTA that relays messages to other ADMDs.

Inbound MTA: An MTA that receives inbound SMTP messages from MTA Relays in other ADMDs, for example, an MTA running on the host listed as the target of an MX record.

Final MTA: The MTA that transfers a message to the MDA.

These identities are relevant to the MTA:

RFC5321.HELO/.EHLO

RFC3461.ENVID

RFC5321.MailFrom

RFC5321.RcptTo

RFC5322.Received: Set by - Relay Server

RFC0791.SourceAddr

4.3.3. Mail Delivery Agent (MDA)

A transfer of responsibility from the MHS to a Recipient's environment (mailbox) is called "delivery". In the architecture, as depicted in Figure 5, delivery takes place within a Mail Delivery Agent (MDA) and is shown as the (D) transition from the MHS-oriented MDA component (hMDA) to the Recipient-oriented MDA component (rMDA).

An MDA can provide distinctive, address-based functionality, made possible by its detailed information about the properties of the destination address. This information might also be present elsewhere in the Recipient's ADMD, such as at an organizational border (Boundary) Relay. However, it is required for the MDA, if only because the MDA is required to know where to deliver the message.

Like an MSA, an MDA serves two roles, as depicted in Figure 5. Formal transfer of responsibility, called "delivery", is effected between the two components that embody these roles and is shown as "(D)" in Figure 5. The MHS portion (hMDA) primarily functions as a server SMTP engine. A common additional role is to redirect the message to an alternative address, as specified by the Recipient addressee's preferences. The job of the Recipient portion of the MDA (rMDA) is to perform any delivery actions that the Recipient specifies.

Transfer into the MDA is accomplished by a normal MTA transfer mechanism. Transfer from an MDA to an MS uses an access protocol, such as POP or IMAP.

NOTE: The term "delivery" can refer to the formal, MHS function specified here or to the first time a message is displayed to a Recipient. A simple, practical test for whether the MHS-based definition applies is whether a DSN can be generated.

These identities are relevant to the MDA:

RFC5321.Return-Path: Set by - Author Originator or Mediator Originator

The MDA records the RFC5321.MailFrom address into the RFC5321.Return-Path field.

RFC5322.Received: Set by - MDA server

An MDA can record a Received: header field to indicate trace information, including source host and receiving host domain names and/or IP Addresses.

4.4. Transition Modes

From the origination site to the point of delivery, Internet Mail usually follows a "push" model. That is, the Actor that holds the message initiates transfer to the next venue, typically with SMTP [RFC5321] or the Local Mail Transfer Protocol (LMTP) [RFC2033]. With a "pull" model, the Actor that holds the message waits for the Actor

in the next venue to initiate a request for transfer. Standardized mechanisms for pull-based MHS transfer are ETRN [RFC1985] and ODMR [RFC2645].

After delivery, the Recipient's MUA (or MS) can gain access by having the message pushed to it or by having the receiver of access pull the message, such as by using POP [RFC1939] and IMAP [RFC3501].

4.5. Implementation and Operation

A discussion of any interesting system architecture often bogs down when architecture and implementation are confused. An architecture defines the conceptual functions of a service, divided into discrete conceptual modules. An implementation of that architecture can combine or separate architectural components, as needed for a particular operational environment. For example, a software system that primarily performs message relaying is an MTA, yet it might also include MDA functionality. That same MTA system might be able to interface with non-Internet email services and thus perform both as an MTA and as a Gateway.

Similarly, implemented modules might be configured to form elaborations of the architecture. An interesting example is a distributed MS. One portion might be a remote server and another might be local to the MUA. As discussed in [RFC1733], there are three operational relationships among such MSs:

Online: The MS is remote, and messages are accessible only when the MUA is attached to the MS so that the MUA will re-fetch all or part of a message from one session to the next.

Offline: The MS is local to the User, and messages are completely moved from any remote store, rather than (also) being retained there.

Disconnected: An rMS and a uMS are kept synchronized, for all or part of their contents, while they are connected. When they are disconnected, mail can arrive at the rMS and the User can make changes to the uMS. The two stores are re-synchronized when they are reconnected.

5. Mediators

Basic message transfer from Author to Recipients is accomplished by using an asynchronous store-and-forward communication infrastructure in a sequence of independent transmissions through some number of MTAs. A very different task is a sequence of postings and deliveries through Mediators. A Mediator forwards a message through a

re-posting process. The Mediator shares some functionality with basic MTA relaying, but has greater flexibility in both addressing and content than is available to MTAs.

This is the core set of message information that is commonly set by all types of Mediators:

RFC5321.HELO/.EHLO: Set by - Mediator Originator

RFC3461.ENVID: Set by - Mediator Originator

RFC5321.RcptTo: Set by - Mediator Author

RFC5321.Received: Set by - Mediator Dest

The Mediator can record received information to indicate the delivery to the original address and submission to the alias address. The trace of Received: header fields can include everything from original posting, through relaying, to final delivery.

The aspect of a Mediator that distinguishes it from any other MUA creating a message is that a Mediator preserves the integrity and tone of the original message, including the essential aspects of its origination information. The Mediator might also add commentary.

Examples of MUA messages that a Mediator does not create include:

New message that forwards an existing message:

Although this action provides a basic template for a class of Mediators, its typical occurrence is not, itself, an example of a Mediator. The new message is viewed as being from the Actor that is doing the forwarding, rather than from the original Author.
A new message encapsulates the original message and is seen as from the new Originator. This Mediator Originator might add commentary and can modify the original message content. Because the forwarded message is a component of the message sent by the new Originator, the new message creates a new dialogue. However, the final Recipient still sees the contained message as from the original Author.

Reply:

When a Recipient responds to the Author of a message, the new message is not typically viewed as a forwarding of the original. Its focus is the new content, although it might

contain all or part of the material from the original message. The earlier material is merely contextual and secondary. This includes automated replies, such as vacation out-of-office notices, as discussed in Section 4.2.1.

Annotation:

The integrity of the original message is usually preserved, but one or more comments about the message are added in a manner that distinguishes commentary from original text. The primary purpose of the new message is to provide commentary from a new Author, similar to a Reply.

The remainder of this section describes common examples of Mediators.

5.1. Alias

One function of an MDA is to determine the internal location of a mailbox in order to perform delivery. An Alias is a simple re-addressing facility that provides one or more new Internet Mail addresses, rather than a single, internal one; the message continues through the transfer service, for delivery to one or more alternate addresses. Although typically implemented as part of an MDA, this facility is a Recipient function. It resubmits the message, although all handling information except the envelope Recipient (rfc5321.RcptTo) address is retained. In particular, the Return Address (rfc5321.MailFrom) is unchanged.

What is distinctive about this forwarding mechanism is how closely it resembles normal MTA store-and-forward relaying. Its only significant difference is that it changes the RFC5321.RcptTo value. Because this change is so small, aliasing can be viewed as a part of the lower-level mail-relaying activity. However, this small change has a large semantic impact: The designated Recipient has chosen a new Recipient.

NOTE: When the replacement list includes more than one address, the alias is increasingly likely to have delivery problems. Any problem reports go to the original Author, not the administrator of the alias entry. This makes it more difficult to resolve the problem, because the original Author has no knowledge of the Alias mechanism.

Including the core set of message information listed at the beginning of this section, Alias typically changes:

RFC5322.To/.CC/.BCC: Set by - Author

These fields retain their original addresses.

RFC5321.MailFrom: Set by - Author

The benefit of retaining the original MailFrom value is to ensure that an Actor related to the originating ADMD knows there has been a delivery problem. On the other hand, the responsibility for handling problems, when transiting from the original Recipient mailbox to the alias mailbox usually lies with that original Recipient, because the Alias mechanism is strictly under that Recipient's control. Retaining the original MailFrom address prevents this.

5.2. ReSender

Also called the ReDirector, the ReSender's actions differ from forwarding because the Mediator "splices" a message's addressing information to connect the Author of the original message with the Recipient of the new message. This connection permits them to have direct exchange, using their normal MUA Reply functions, while also recording full reference information about the Recipient who served as a Mediator. Hence, the new Recipient sees the message as being from the original Author, even if the Mediator adds commentary.

Including the core set of message information listed at the beginning of this section, these identities are relevant to a resent message:

RFC5322.From: Set by - original Author

Names and addresses for the original Author of the message content are retained. The free-form (display-name) portion of the address might be modified to provide an informal reference to the ReSender.

RFC5322.Reply-To: Set by - original Author

If this field is present in the original message, it is retained in the resent message.

RFC5322.Sender: Set by - Author's Originator or Mediator Originator

RFC5322.To/.CC/.BCC: Set by - original Author

These fields specify the original message Recipients.

RFC5322.Resent-From: Set by - Mediator Author

This address is of the original Recipient who is redirecting the message. Otherwise, the same rules apply to the Resent-From: field as to an original RFC5322.From field.

RFC5322.Resent-Sender: Set by - Mediator Originator

The address of the Actor responsible for resubmitting the message. As with RFC5322.Sender, this field can be omitted when it contains the same address as RFC5322.Resent-From.

RFC5322.Resent-To/-CC/-BCC: Set by - Mediator Author

The addresses of the new Recipients who are now able to reply to the original Author.

RFC5321.MailFrom: Set by - Mediator Originator

The Actor responsible for resubmission (RFC5322.Resent-Sender) is also responsible for specifying the new MailFrom address.

5.3. Mailing Lists

A Mailing List receives messages as an explicit addressee and then re-posts them to a list of subscribed members. The Mailing List performs a task that can be viewed as an elaboration of the ReSender. In addition to sending the new message to a potentially large number of new Recipients, the Mailing List can modify content, for example, by deleting attachments, converting the format, and adding list-specific comments. Mailing Lists also archive messages posted by Authors. Still the message retains characteristics of being from the original Author.

Including the core set of message information listed at the beginning of this section, these identities are relevant to a Mailing List processor when submitting a message:

RFC2919.List-Id: Set by - Mediator Author

RFC2369.List-*: Set by - Mediator Author

RFC5322.From: Set by - original Author

Names and email addresses for the original Author of the message content are retained.

RFC5322.Reply-To: Set by - Mediator or original Author

Although problematic, it is common for a Mailing List to assign its own addresses to the Reply-To: header field of messages that it posts. This assignment is intended to ensure that replies go to all list members, rather than to only the original Author. As a User Actor, a Mailing List is the Author of the new message and can legitimately set the Reply-To: value. As a Mediator attempting to represent the message on behalf of its original Author, creating or modifying a Reply-To: field can be viewed as violating that Author's intent. When the Reply-To is modified in this way, a reply that is meant only for the original Author will instead go to the entire list. When the Mailing List does not set the field, a reply meant for the entire list can instead go only to the original Author. At best, either choice is a matter of group culture for the particular list.

RFC5322.Sender: Set by - Author Originator or Mediator Originator

This field usually specifies the address of the Actor responsible for Mailing List operations. Mailing Lists that operate in a manner similar to a simple MTA Relay preserve as much of the original handling information as possible, including the original RFC5322.Sender field. (Note that this mode of operation causes the Mailing List to behave much like an Alias, with a possible difference in number of new addressees.)

RFC5322.To/.CC: Set by - original Author

These fields usually contain the original list of Recipient addresses.

RFC5321.MailFrom: Set by - Mediator Originator

Because a Mailing List can modify the content of a message in any way, it is responsible for that content; that is, it is an Author. As such, the Return Address is specified by the Mailing List. Although it is plausible for the Mailing List to reuse the Return Address employed by the original Originator, notifications sent to that address after a message has been processed by a Mailing List could be problematic.

5.4. Gateways

A Gateway performs the basic routing and transfer work of message relaying, but it also is permitted to modify content, structure, address, or attributes as needed to send the message into a messaging environment that operates under different standards or potentially incompatible policies. When a Gateway connects two differing messaging services, its role is easy to identify and understand. When it connects environments that follow similar technical standards, but significantly different administrative policies, it is easy to view a Gateway as merely an MTA.

The critical distinction between an MTA and a Gateway is that a Gateway can make substantive changes to a message to map between the standards. In virtually all cases, this mapping results in some degree of semantic loss. The challenge of Gateway design is to minimize this loss. Standardized Gateways to Internet Mail are facsimile [RFC4143], voicemail [RFC3801], and the Multimedia Messaging Service (MMS) [RFC4356].

A Gateway can set any identity field available to an MUA. Including the core set of message information listed at the beginning of this section, these identities are typically relevant to Gateways:

RFC5322.From: Set by - original Author

Names and addresses for the original Author of the message content are retained. As for all original addressing information in the message, the Gateway can translate addresses as required to continue to be useful in the target environment.

RFC5322.Reply-To: Set by - original Author

It is best for a Gateway to retain this information, if it is present. The ability to perform a successful reply by a Recipient is a typical test of Gateway functionality.

RFC5322.Sender: Set by - Author Originator or Mediator Originator

This field can retain the original value or can be set to a new address.

RFC5322.To/.CC/.BCC: Set by - original Recipient

These fields usually retain their original addresses.

RFC5321.MailFrom: Set by - Author Originator or Mediator Originator

The Actor responsible for handling the message can specify a new address to receive handling notices.

5.5. Boundary Filter

To enforce security boundaries, organizations can subject messages to analysis for conformance with its safety policies. An example is detection of content classed as spam or a virus. A filter might alter the content to render it safe, such as by removing content deemed unacceptable. Typically, these actions add content to the message that records the actions.

6. Considerations

6.1. Security Considerations

This document describes the existing Internet Mail architecture. It introduces no new capabilities. The security considerations of this deployed architecture are documented extensively in the technical specifications referenced by this document. These specifications cover classic security topics, such as authentication and privacy. For example, email-transfer protocols can use standardized mechanisms for operation over authenticated and/or encrypted links, and message content has similar protection standards available. Examples of such mechanisms include SMTP-TLS [RFC3207], SMTP-Auth [RFC4954], OpenPGP [RFC4880], and S/MIME [RFC3851].

The core of the Internet Mail architecture does not impose any security requirements or functions on the end-to-end or hop-by-hop components. For example, it does not require participant authentication and does not attempt to prevent data disclosure.

Particular message attributes might expose specific security considerations. For example, the blind carbon copy feature of the architecture invites disclosure concerns, as discussed in Section 7.2 of [RFC5321] and Section 5 of [RFC5322]. Transport of text or non-text content in this architecture has security considerations that are discussed in [RFC5322], [RFC2045], [RFC2046], and [RFC4288]; also, security considerations are present for some of the media types registered with IANA.

Agents that automatically respond to email raise significant security considerations, as discussed in [RFC3834]. Gateway behaviors affect end-to-end security services, as discussed in [RFC2480]. Security considerations for boundary filters are discussed in [RFC5228].

See Section 7.1 of [RFC5321] for a discussion of the topic of origination validation. As mentioned in Section 4.1.4, it is common practice for components of this architecture to use the RFC0791.SourceAddr to make policy decisions [RFC2505], although the address can be "spoofed". It is possible to use it without authorization. SMTP and Submission authentication ([RFC4409], [RFC4954]) provide more secure alternatives.

The discussion of trust boundaries, ADMDs, Actors, roles, and responsibilities in this document highlights the relevance and potential complexity of security factors for operation of an Internet Mail service. The core design of Internet Mail to encourage open and casual exchange of messages has met with scaling challenges, as the population of email participants has grown to include those with problematic practices. For example, spam, as defined in [RFC2505], is a by-product of this architecture. A number of Standards Track or BCP documents on the subject have been issued (see [RFC2505], [RFC5068], and [RFC5235]).

6.2. Internationalization

The core Internet email standards are based on the use of US-ASCII -- that is, SMTP [RFC5321] and IMF [RFC5322], as well as their predecessors. They describe the transport and composition of messages as composed strictly of US-ASCII 7-bit encoded characters. The standards have been incrementally enhanced to allow for characters outside of this limited set, while retaining mechanisms for backwards-compatibility. Specifically:

- The MIME specifications ([RFC2045], [RFC2046], [RFC2047], [RFC2049]) allow for the use of coded character sets and character-encoding schemes ("charsets" in MIME terminology) other than US-ASCII. MIME's [RFC2046] allows the textual content of a message to have a label affixed that specifies the charset used in that content. Equally, MIME's [RFC2047] allows the textual content of certain header fields in a message to be similarly labeled. However, since messages might be transported over SMTP implementations only capable of transporting 7-bit encoded characters, MIME's [RFC2045] also provides for "content transfer encoding" so that characters of other charsets can be re-encoded as an overlay to US-ASCII.

- MIME's [RFC2045] allows for the textual content of a message to be in an 8-bit character-encoding scheme. In order to transport these without re-encoding them, the SMTP specification supports an option [RFC1652] that permits the transport of such textual

content. However, the [RFC1652] option does not address the use of 8-bit content in message header fields, and therefore [RFC2047] encoding is still required for those.

- A series of experimental protocols on Email Address Internationalization (EAI) have been released that extend SMTP and IMF to allow for 8-bit encoded characters to appear in addresses and other information throughout the header fields of messages. [RFC5335] specifies the format of such message header fields (which encode the characters in UTF-8), and [RFC5336] specifies an SMTP option for the transport of these messages.

- MIME's [RFC2045] and [RFC2046] allow for the transport of true multimedia material; such material enables internationalization because it is not restricted to any particular language or locale.

- The formats for Delivery Status Notifications (DSNs -- [RFC3462], [RFC3463], [RFC3464]) and Message Disposition Notifications (MDNs -- [RFC3798]) include both a structured and unstructured representation of the notification. In the event that the unstructured representation is in the wrong language or is otherwise unsuitable for use, this allows an MUA to construct its own appropriately localized representation of notification for display to the User.

- POP and IMAP have no difficulties with handling MIME messages, including ones containing 8bit, and therefore are not a source of internationalization issues.

Hence, the use of UTF-8 is fully established in existing Internet Mail. However, support for long-standing encoding forms is retained and is still used.

7. References

7.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2369] Neufeld, G. and J. Baer, "The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields", RFC 2369, July 1998.

[RFC2645] Gellens, R., "ON-DEMAND MAIL RELAY (ODMR) SMTP with Dynamic IP Addresses", RFC 2645, August 1999.

[RFC2919] Chandhok, R. and G. Wenger, "List-Id: A Structured Field and Namespace for the Identification of Mailing Lists", RFC 2919, March 2001.

[RFC3192] Allocchio, C., "Minimal FAX address format in Internet Mail", RFC 3192, October 2001.

[RFC3297] Klyne, G., Iwazaki, R., and D. Crocker, "Content Negotiation for Messaging Services based on Email", RFC 3297, July 2002.

[RFC3458] Burger, E., Candell, E., Eliot, C., and G. Klyne, "Message Context for Internet Mail", RFC 3458, January 2003.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[RFC3462] Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 3462, January 2003.

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3798] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[RFC3834] Moore, K., "Recommendations for Automatic Responses to Electronic Mail", RFC 3834, August 2004.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, March 2005.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4289] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 4289, December 2005.

[RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[RFC4550] Maes, S. and A. Melnikov, "Internet Email to Support Diverse Service Environments (Lemonade) Profile", RFC 4550, June 2006.

[RFC5228] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", RFC 5228, January 2008.

[RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, June 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

7.2. Informative References

[RFC0733] Crocker, D., Vittal, J., Pogran, K., and D. Henderson, "Standard for the format of ARPA network text messages", RFC 733, November 1977.

[RFC0821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[RFC1506] Houttuin, J., "A Tutorial on Gatewaying between X.400 and Internet Mail", RFC 1506, August 1993.

[RFC1652] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

[RFC1733] Crispin, M., "Distributed Electronic Mail Models in IMAP4", RFC 1733, December 1994.

[RFC1767] Crocker, D., "MIME Encapsulation of EDI Objects", RFC 1767, March 1995.

[RFC1985] De Winter, J., "SMTP Service Extension for Remote Message Queue Starting", RFC 1985, August 1996.

[RFC2033] Myers, J., "Local Mail Transfer Protocol", RFC 2033, October 1996.

[RFC2142] Crocker, D., "MAILBOX NAMES FOR COMMON SERVICES, ROLES AND FUNCTIONS", RFC 2142, May 1997.

[RFC2442] Freed, N., Newman, D., and Hoy, M., "The Batch SMTP Media Type", RFC 2442, November 1998.

[RFC2480] Freed, N., "Gateways and MIME Security Multiparts", RFC 2480, January 1999.

[RFC2505] Lindberg, G., "Anti-Spam Recommendations for SMTP MTAs", BCP 30, RFC 2505, February 1999.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[RFC3801] Vaudreuil, G. and G. Parsons, "Voice Profile for Internet Mail - version 2 (VPIMv2)", RFC 3801, June 2004.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[RFC3885] Allman, E. and T. Hansen, "SMTP Service Extension for Message Tracking", RFC 3885, September 2004.

[RFC4142] Crocker, D. and G. Klyne, "Full-mode Fax Profile for Internet Mail (FFPIM)", RFC 4142, November 2005.

[RFC4143] Toyoda, K. and D. Crocker, "Facsimile Using Internet Mail (IFAX) Service of ENUM", RFC 4143, November 2005.

[RFC4356] Gellens, R., "Mapping Between the Multimedia Messaging Service (MMS) and Internet Mail", RFC 4356, January 2006.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RFC4954] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[RFC5068] Hutzler, C., Crocker, D., Resnick, P., Allman, E., and T. Finch, "Email Submission Operations: Access and Accountability Requirements", BCP 134, RFC 5068, November 2007.

[RFC5235] Daboo, C., "Sieve Email Filtering: Spamtest and Virustest Extensions", RFC 5235, January 2008.

[RFC5335] Abel, Y., "Internationalized Email Headers", RFC 5335, September 2008.

[RFC5336] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email Addresses", RFC 5336, September 2008.

[Tussle] Clark, D., Wroclawski, J., Sollins, K., and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet", ACM SIGCOMM, 2002.

Appendix A. Acknowledgments

This work began in 2004 and has evolved through numerous rounds of community review; it derives from a section in an early version of [RFC5068]. Over its 5 years of development, the document has gone through 14 incremental versions, with vigorous community review that produced many substantive changes. Review was performed in the IETF and other email technical venues. Although not a formal activity of the IETF, issues with the document's contents were resolved using the classic style of IETF community open, group decision-making. The document is already cited in other work, such as in IMAP and Sieve specifications and in academic classwork. The step of standardizing is useful to provide a solid and stable reference to the Internet's now-complex email service.

Details of the Originator Actor role was greatly clarified during discussions in the IETF's Marid working group.

Graham Klyne, Pete Resnick, and Steve Atkins provided thoughtful insight on the framework and details of the original drafts, as did Chris Newman for the final versions, while also serving as cognizant Area Director for the document. Tony Hansen served as document shepherd through the IETF process.

Later reviews and suggestions were provided by Eric Allman, Nathaniel Borenstein, Ed Bradford, Cyrus Daboo, Frank Ellermann, Tony Finch, Ned Freed, Eric Hall, Willemien Hoogendoorn, Brad Knowles, John Leslie, Bruce Valdis Kletnieks, Mark E. Mallett, David MacQuigg, Alexey Melnikov, der Mouse, S. Moonesamy, Daryl Odnert, Rahmat M. Samik-Ibrahim, Marshall Rose, Hector Santos, Jochen Topf, Greg Vaudreuil, Patrick Cain, Paul Hoffman, Vijay Gurbani, and Hans Lachman.

Diligent early proof-reading was performed by Bruce Lilly. Diligent professional technical editing was provided by Susan Hunziker.

The final stages of development for this document were guided by a design team comprising Alexey Melnikov, Pete Resnick, Carl S. Gutekunst, Jeff Macdonald, Randall Gellens, Tony Hansen, and Tony Finch. Pete Resnick developed the final version of the section on internationalization.

Index

Author's Address

Dave Crocker
Brandenburg InternetWorking
675 Spruce Drive
Sunnyvale, CA 94086
USA

Phone: +1.408.246.8253
EMail: dcrocker@bbiw.net