

SMTP
Internet-Draft
Intended status: Standards Track
Expires: November 7, 2007

D. Crocker
Brandenburg InternetWorking
May 6, 2007

Internet Mail Architecture
draft-crocker-email-arch-07

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on November 7, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Over its thirty-five year history Internet Mail has undergone significant changes in scale and complexity, as it has become a global infrastructure service. The first standardized architecture for networked email specified little more than a simple split between the user world and the transmission world. Core aspects of the service, such as the styles of mailbox address and basic message format, have remained remarkably constant. However today's Internet Mail is marked by many independent operators, many different

Internet-Draft

EMail Architecture

May 2007

components for providing service to users and many others for performing message transfer. Public discussion of the service often lacks common terminology and a common frame of reference for these components and their activities. Having a common reference model and terminology makes a basic difference when talking about problems with the service, changes in policy, or enhancement to the service's functionality. This document offers an enhanced Internet Mail architecture that targets description of the existing service, in order to facilitate clearer and more efficient technical, operations and policy discussions about email.

Table of Contents

1.	Introduction	3
1.1.	Background	3
1.2.	Service Overview	5
1.3.	Document Conventions	6
2.	Responsible Actor Roles	6
2.1.	User Actors	7
2.2.	Mail Handling Service (MHS) Actors	10
2.3.	Administrative Actors	13
3.	Identities	15
3.1.	Mailbox	15
3.2.	Domain Names	16
3.3.	Message Identifier	17
4.	Services and Standards	18
4.1.	Message Data	21
4.2.	User-Level Services	26
4.3.	MHS-Level Services	28
5.	Mediators	31
5.1.	Aliasing	32
5.2.	Re-Sending	34
5.3.	Mailing Lists	36
5.4.	Gateways	37
5.5.	Boundary Filter	39
6.	Security Considerations	39
7.	IANA Considerations	39
8.	References	39
8.1.	Normative	39
8.2.	Descriptive	42
Appendix A.	Acknowledgements	42
	Author's Address	43

Crocker

Expires November 7, 2007

[Page 2]

Internet-Draft

EMail Architecture

May 2007

[1.](#) Introduction

Over its thirty-five year history Internet Mail has undergone significant changes in scale and complexity, as it has become a global infrastructure service. The changes have been evolutionary, rather than revolutionary, reflecting a strong desire to preserve its installed base of users and utility. Today, Internet Mail is marked by many independent operators, many different components for providing service to users and many other components for performing message transfer.

Public collaboration on email technical, operations and policy activities, including those responding to the challenges of email abuse, has brought in a much wider range of participants than email's technical community originally had. In order to do work on a large, complex system, they need to share the same view of how it is put together, as well as what terms to use to refer to the pieces and their activities. Otherwise, it is difficult to know exactly what another participant means. It is these differences in each person's perspective that motivates this document, to describe 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. This architecture document seeks to facilitate clearer and more efficient technical, operations and policy exchanges about email.

This document offers an enhanced Internet Mail architecture to reflect the current service. In particular it:

- * Documents refinements to the email model
- * Clarifies functional roles for the architectural components

- ### 1.1. Background

Crocker

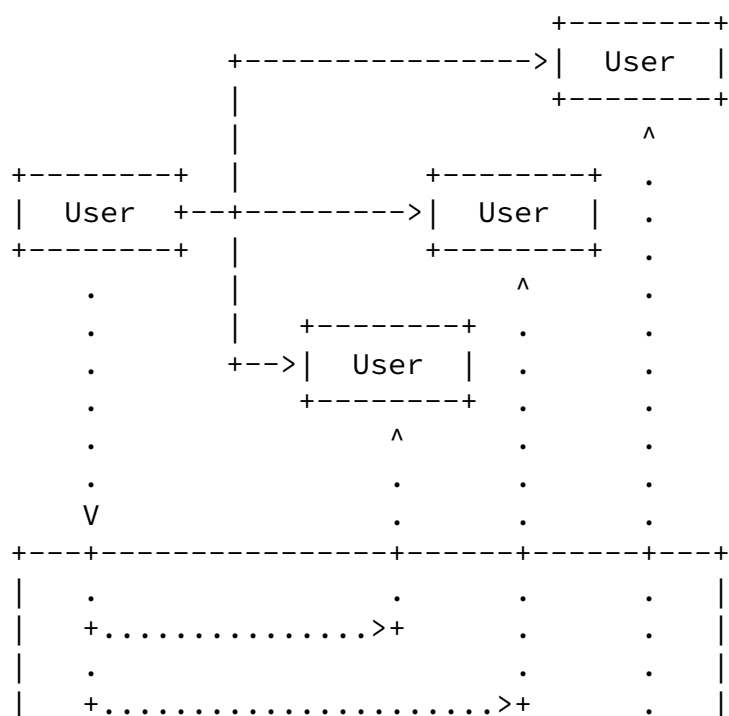
[Page 3]

Internet-Draft

EMail Architecture

May 2007

As shown in Figure 1 this defines two logical "layers" of interoperability. One is directly between Users. The other is between the neighboring 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.



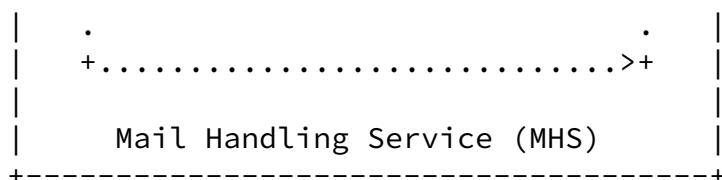


Figure 1: Basic Internet Mail Service Model

As it has evolved, the operational service has sub-divided each of these layers into more specialized modules. Core aspects of the service, such as mailbox addressing and message format style, have remained remarkably constant. So the original distinction between user-level concerns and transfer-level concerns is retained, but with an elaboration to each level of the architecture. 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 directly resulting from its single

transiting of the MHS. A common exception is with group dialogue that is mediated via a mailing list, so that two postings occur before intended recipients receive an originator's message, as discussed in [Section 2.1.3](#). In fact some uses of email consider the entire email service -- including Originator and Recipient -- as a subordinate component. For these services "end-to-end" refers to points outside of the email service. Examples are voicemail over email [[RFC3801](#)], EDI over email [[RFC1767](#)] and facsimile over email. [[RFC4142](#)]

[1.2](#). Service Overview

End-to-end Internet Mail exchange is accomplished by using a standardized infrastructure comprising:

- * An email object
- * Global addressing
- * An asynchronous sequence of point-to-point transfer mechanisms

- * No prior arrangement between Originator and Recipient
- * No prior arrangement between point-to-point transfer services, over the open Internet
- * No requirement for Originator and Recipient 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 between control information for handling, versus the author's message content.

A precept to the design of mail over the open Internet is permitting user-to-user and MTA-to-MTA interoperability to take place with no prior, direct arrangement between the independent administrative authorities that are responsible for handling a message. That is, all participants rely on having the core services be universally supported and accessible, either directly or through gateways that translate between Internet Mail standards and other email environments. Given the importance of spontaneity and serendipity in the world of human communications, this lack of 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 lookup service configuration. In recent years one change to local environments is an increased requirement for authentication or, at least, accountability. In these cases a server performs explicit validation of the client's identity.

1.3. Document Conventions

In this document, 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 is the name of the field. Hence <[RFC2822](#).From> is the From field in an email content header and <[RFC2821](#).MailFrom> is the address in the SMTP "Mail From" command.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as specified in [RFC 2119](http://www.imc.org/rfc2119) [[RFC2119](http://www.imc.org/rfc2119)].

Discussion venue: Please direct discussion about this document to the IETF-SMTP mailing list <<http://www.imc.org/ietf-smtp>>.

Changes:

Added text to explain utility of having an architecture document.

Added text explaining benefit of the ADMD construct.

Added commentary on List-ID.

Moved Bounce out of MHS in figure.

Moved "generic" Identity field case-analysis text into common area after Table 1, reserving per-role text for per-role peculiarities.

Extensive word-smithing and cleanup.

[2.](#) Responsible Actor Roles

Internet Mail is a highly distributed service, with a variety of actors serving different roles. These divide into 3 basic types:

- * User

Crocker

Expires November 7, 2007

[Page 6]

Internet-Draft

EMail Architecture

May 2007

- * Mail Handling Service (MHS)

- * ADministrative Management Domain (ADMD)

Although related to a technical architecture, the focus on Actors concerns participant responsibilities, rather than on functionality of modules. Hence the labels used are different than for classic email architecture diagrams.

[2.1.](#) User Actors

Users are the sources and sinks of messages. They can be humans or processes. They can have an exchange that iterates and they can expand or contract the set of users participating in a set of exchanges. In Internet Mail there are three types of user-level Actors:

- * Originators
- * Recipients
- * Mediators

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

The following depicts the flow of messages among Actors:

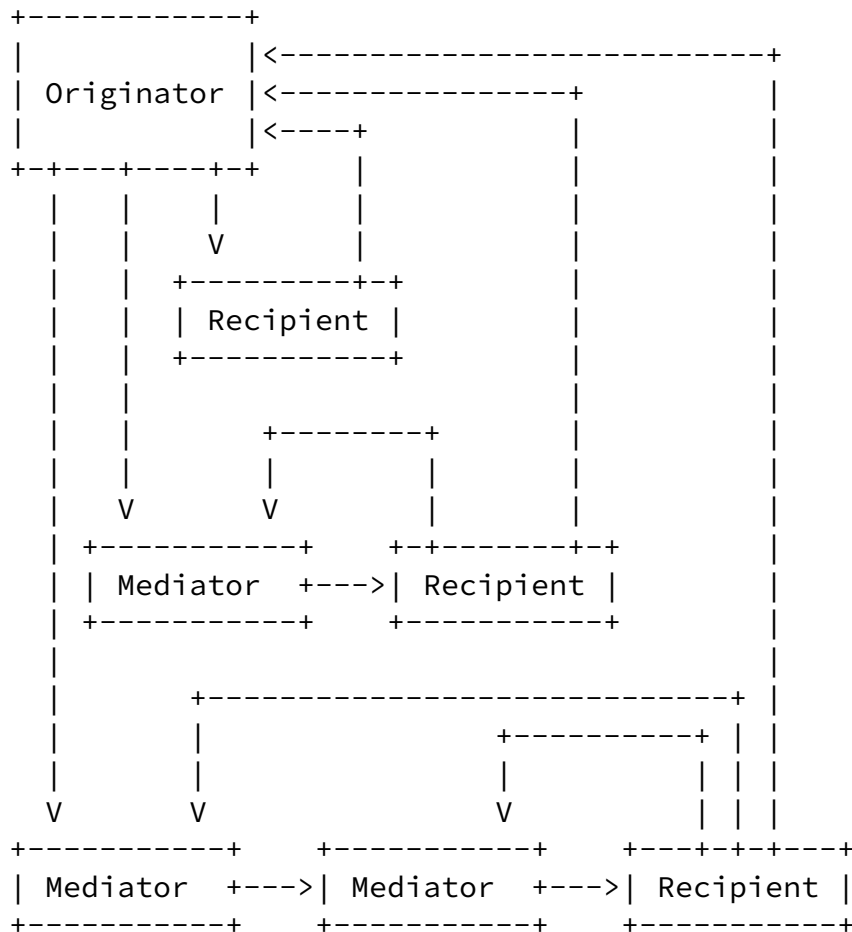


Figure 2: Relationships Among User Actors

[2.1.1.](#) Originator

Also called "Author", this is the user-level participant responsible for creating original content and requesting its transmission. The MHS operates to send and deliver mail among Originators and Recipients. As described below, the MHS has a "Source" role that correlates with the user-level Author role.

[2.1.2.](#) Recipient

The Recipient is a consumer of delivered content. As described below, the MHS has a "Dest[ination]" role that correlates with the user-level Recipient role.

A Recipient can close the user-level communication loop by creating and submitting a new message that replies to an Originator. An example of an automated form of reply is the Message Disposition Notification, which informs the Originator about the Recipient's

handling of the message. (See [Section 4.1.](#))

[2.1.3.](#) Mediator

A Mediator receives, aggregates, reformulates and redistributes messages as part of a potentially-protracted, higher-level exchange among Users. Example Mediators include group dialogue, such as collaboration via mailing lists, and organizational message flow, as occurs with a purchase approval process. Note that it is easy to confuse this user-level activity with the underlying MHS transfer exchanges. However they serve very different purposes and operate in very different ways. Mediators are considered extensively in [Section 5.](#)

When mail is delivered to a receiving mediator specified in the [RFC2821](#).RcptTo command, the MHS handles it the same way as for any other Recipient. That is, the MHS only sees posting and delivery sources and sinks and does not see (later) re-posting as a continuation of a process. Hence when submitting messages, the Mediator is an Originator.

The distinctive aspects of a Mediator are, therefore, above the MHS. A Mediator preserves the Originator information of the message it reformulates, but may make meaningful changes to the content. Hence the MHS sees a new message, but Users receive a message that is interpreted as primarily being from -- or, at least, initiated by -- the author of the original message. The role of a Mediator permits distinct, active creativity, rather than being limited to the more constrained job of merely connecting together other participants. Hence it is really the Mediator that is responsible for the new message.

A Mediator's task can be complex and contingent, such as by modifying and adding content or regulating which users are allowed to participate and when. The popular example of this role is a group mailing list. A sequence of Mediators may even perform a series of formal steps, such as reviewing, modifying and approving a purchase request.

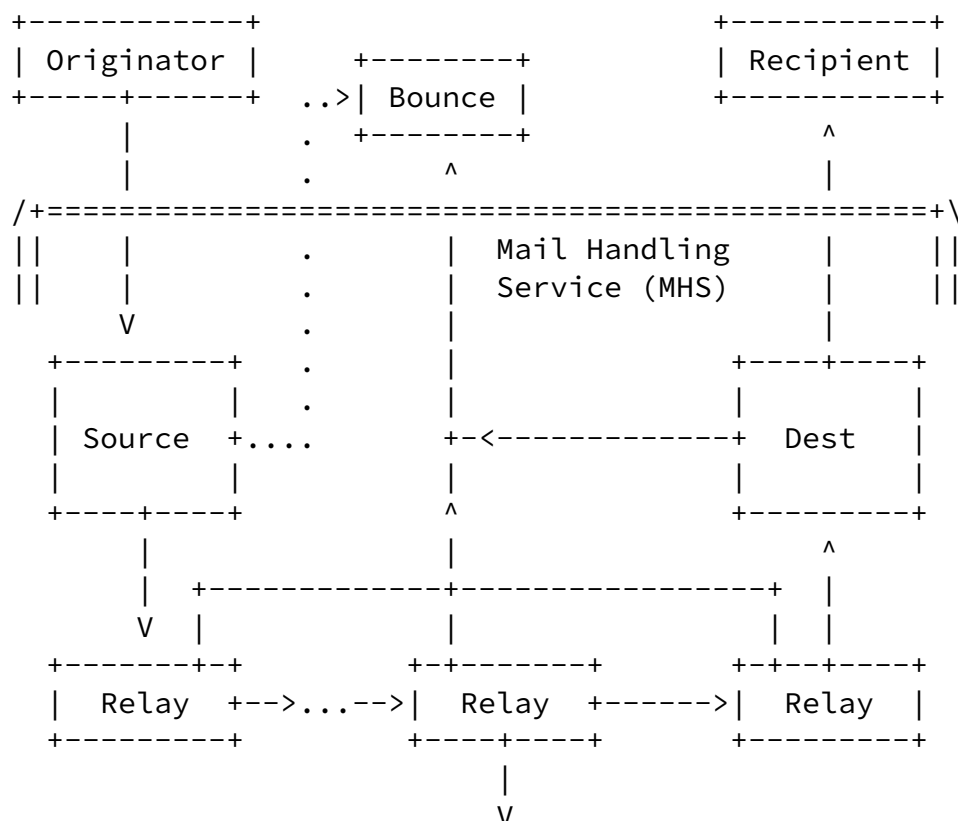
Because a Mediator originates messages, it can also receive replies. So a Mediator really is a full-fledged User.

Gateway: A Gateway is a particularly interesting form of Mediator. It is a hybrid of User and Relay that interconnects heterogeneous mail services. Its goal is to emulate a Relay, and a detailed discussion is in [Section 2.2.4.](#)

2.2. Mail Handling Service (MHS) Actors

The Mail Handling Service (MHS) has the task of performing a single, end-to-end transfer on behalf of the Originator and reaching the Recipient address(es) specified in the original [RFC2821](#).RcptTo commands. Mediated or protracted, iterative exchanges, such as those used for collaboration over time, are part of the User-level service, and are not part of this transfer-level Handling Service.

The following depicts the relationships among transfer participants in Internet Mail. It shows the Source as distinct from the Originator, and Dest[ination] as distinct from Recipient, although it is common for each pair to be the same actor. Transfers typically entail one or more Relays. However direct delivery from the Source to Destination is possible. For intra-organization mail services, it is common to have only one Relay.



```
+-----+
| Gateway +-->...
+-----+
```

Figure 3: Relationships Among MHS Actors

Crocker

Expires November 7, 2007

[Page 10]

Internet-Draft

E-Mail Architecture

May 2007

[2.2.1.](#) Source

The Source role is responsible for ensuring that a message is valid for posting and then submitting it to a Relay. Validity includes conformance with Internet Mail standards, as well as with local operational policies. The Source can simply review the message for conformance and reject it if there are errors, or it can create some or all of the necessary information.

The Source operates with dual "allegiance". It serves the Originator and often it is the same entity. However its role in assuring validity means that it MUST also represent the local operator of the MHS, that is, the local Administrative Management Domain (ADMD).

The Source also has the responsibility for any post-submission, Originator-related administrative tasks associated with message transmission and delivery. Notably this pertains to error and delivery notices. Hence Source is best held accountable for the message content, even when they did not create any or most of it.

[2.2.2.](#) Bounce Handler

The Bounce Handler processes service notifications that are generated by the MHS, as a result of its efforts to transfer or deliver the message. Notices can be about failures or completions and are sent to an address that is specified by the Source. This Bounce handling address (also known as a Return address) might have no visible characteristics in common with the address of the Originator or Source.

NOTE:

The choice of the label "Bounce" is unfortunate, due to its negative implication and narrow focus. However it is the most popular term for the address.

[2.2.3.](#) Relay

A mail Relay performs email transfer-service routing and store-and-forward by (re-)transmitting the message on towards its Recipient(s). A Relay can add information to the envelope, such as with trace information. However it does not modify existing envelope information or the message content semantics. It can modify message content syntax, such as a change from text to binary transfer-encoding form, only as required to meet the capabilities of the next hop in the MHS.

A set of Relays composes a Mail Handling Service (MHS) network. This

is above any underlying packet-switching network that they might be using and below any gateways or other user-level Mediators.

In other words, interesting email scenarios can involve three distinct architectural layers of store-and-forward service:

- * User Mediators
- * MHS Relays
- * Packet Switches

with the bottom-most usually being the Internet's IP service. The most basic email scenarios involve Relays and Switches.

Aborting a message transfer results in having the Relay become an Originator and send an error message to the Bounce address. The potential for looping is avoided by having this message, itself, contain no Bounce address.

[2.2.4.](#) Gateway

A Gateway is a hybrid form of User and Relay that interconnects

heterogeneous mail services. Its purpose is simply to emulate a Relay and the closer it comes to this, the better. However it operates at the User level, because it MUST be able to modify message content.

Differences between mail services can be as small as minor syntax variations, but usually encompass significant, semantic distinctions. One difference could have the concept of an email address be a hierarchical, machine-specific address, versus having it be a flat, global name space. Another difference could be between text-only content, versus multi-media. Hence the Relay function in a Gateway offers significant design challenges, to make the result be as seamless as possible. The most significant challenge is in ensuring the user-to-user functionality that matches syntax and semantics of independent email standards suites.

The basic test of a Gateway's adequacy is, of course, whether an Originator on one side of a Gateway can send a useful message to a Recipient on the other side, without requiring changes to any of the components in the Originator's or Recipient's mail services, other than adding the Gateway. To each of these otherwise independent services, the Gateway will appear to be a "native" participant. However the ultimate test of a Gateway's adequacy is whether the

Originator and Recipient can sustain a dialogue. In particular can a Recipient's MUA automatically formulate a valid Reply that will reach the initial Originator?

[2.3.](#) Administrative Actors

Actors often will 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 for distinguishing among Administrative Management Domains (ADMD). Each ADMD can have vastly different operating policies and trust-based decision-making. An obvious example is the distinction between mail that is exchanged within a single organization, versus mail that is exchanged between independent organizations. The rules for handling these two 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 apportioned to 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. Examples include an end-user operating their desktop client, a department operating a local Relay, an IT department operating an enterprise Relay and an ISP operating a public shared email service. These can be configured into many combinations of administrative and operational relationships, with each ADMD potentially having a complex arrangement of functional components. Figure 4 depicts relationships among ADMDs. The benefit of having the ADMD construct is to facilitate discussions and designs that need to distinguish between "internal" issues and "external" ones.

The architectural impact of needing to have boundaries between ADMD's is discussed in [[Tussle](#)]. Most significant is that the entities communicating across ADMD boundaries will typically have an added burden to enforce organizational policies concerning "external" communications. At a more mundane level, the basis for routing mail between ADMDs is often an issue.

Basic types of ADMDs include --

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

User: End-user services. This might be subsumed under the Edge service, such as is common for web-based email access.

Transit: These are Mail Service Providers (MSP) offering value-added capabilities for Edge ADMDs, such as aggregation and filtering.

Note that Transit services are quite different from packet-level switching operation. Whereas end-to-end packet transfers usually go through intermediate routers, email exchange across the open Internet is often directly between the Boundary MTAs of Edge ADMDs, at the

email level.

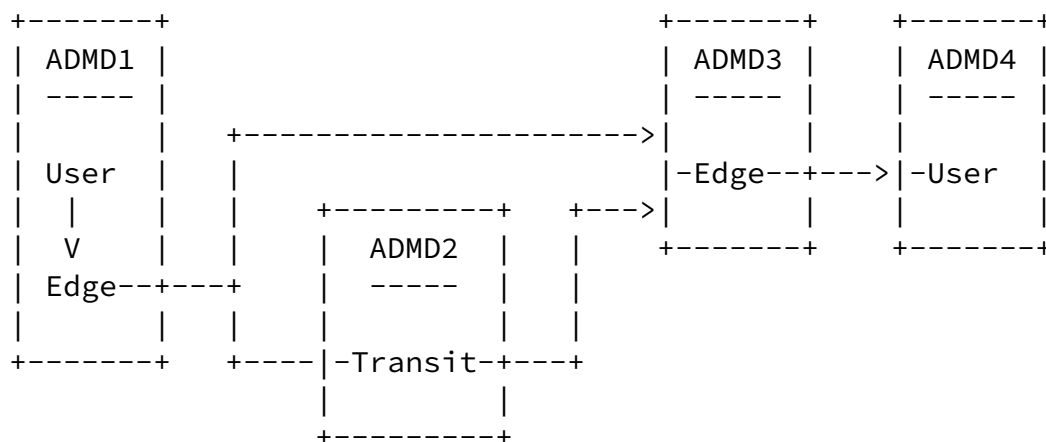


Figure 4: ADministrative Management Domains (ADMD) Example

Edge networks can use proprietary email standards internally. However the distinction between Transit network and Edge network transfer services is primarily 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 transitions of responsibility, as well as the accountability and authorization relationships among participants in email transfer.

The interactions between functional components within an ADMD are subject to the policies of that domain. Policies can cover such things as reliability, access control, accountability and even content evaluation and modification. They can be implemented in different functional components, according to the needs of the ADMD. For example see [[ID-spamops](#)].

User, 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.

Common ADMD examples are --

Enterprise Service Providers:

Operating an organization's internal data and/or mail services.

Internet Service Providers:

Operating underlying data communication services that, in turn, are used by one or more Relays and Users. It is not necessarily their job to perform email functions, but they can, instead, provide an environment in which those functions can be performed.

Mail Service Providers:

Operating email services, such as for end-users, or mailing lists.

Operational pragmatics often dictate that providers be involved in detailed administration and enforcement issues, to help ensure the health of the overall Internet Mail Service. This can include operators of lower-level packet services.

[3.](#) Identities

Internet Mail uses three forms of identity: mailbox, domain name and message-id. Each is required to be globally unique.

[3.1.](#) Mailbox

"A mailbox sends and receives mail. It is a conceptual entity which does not necessarily pertain to file storage." [[RFC2822](#)]

A mailbox is specified as an Internet Mail address <addr-spec>. It has two distinct parts, divided by an at-sign ("@"). The right-hand side is a globally interpreted domain name that is part of an ADMD. Domain Names are discussed in [Section 3.2](#). Formal Internet Mail addressing syntax can support source routes, to indicate the path through which a message should be sent. Although legal, the use of source routes is not part of the modern Internet Mail service and it is ignored in the rest of this document.

The portion to the left of the at-sign contains a string that is globally opaque and is called the <local-part>. It is to be

interpreted only by the entity specified by the address's right-hand side domain name. All other entities **MUST** treat the local-part as a uninterpreted literal string and **MUST** 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 originator.

[3.1.1.](#) Global Standards for Local-Part

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 **MUST** not be interpreted by any domain except the one listed in the right-hand side of the addr-spec, and those specialized services conforming to standardized conventions, as noted in the next paragraph.

There are a few types of addresses that have an elaboration on basic email addressing, with a standardized, global schema for the local-part. These are conventions between originating end-systems and Recipient Gateways, and they are invisible to the public email transfer infrastructure. When an Originator is explicitly sending via a Gateway out of the Internet, there are coding conventions for the local-part, so that the Originator can formulate instructions for the Gateway. Standardized examples of this are the telephone numbering formats for VPIM [[RFC3801](#)], such as "+16137637582@vpim.example.com", and iFax [[RFC3192](#)], such as "FAX=+12027653000/T33S=1387@ifax.example.com".

[3.1.2.](#) Scope of Email Address Use

Email addresses are being used far beyond their original email transfer and delivery role. In practical terms, email strings have become a common form of user identity on the Internet. What is essential, then, is to be clear about the nature and role of an identity string in a particular context and to be clear about the entity responsible for setting that string.

[3.2.](#) 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 might encompass an entire organization, a collection of machines integrated into a homogeneous service, or only a single machine. A domain name can be administered

to refer to individual users, but this is not common practice. The

name is structured as a hierarchical sequence of sub-names, separated by dots ("."), with the top of the hierarchy being on the right-end of the sequence. Domain names are defined and operated through the Domain Name Service (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 the host that took action upon the message, such as providing the administrative scope for a message identifier, or performing transfer processing.

[3.3.](#) Message Identifier

There are two standardized tags, for identifying messages: Message-ID and ENVID.

[3.3.1.](#) Message-ID

The Message-ID is a user-level tag, primarily used for threading and for eliminating duplicates. [[RFC2822](#)]. It is associated with the [RFC2822](#).From field, although any actor within the originating ADMD might assign it. The recipient's ADMD is the intended consumer of the Message-ID, although any actor along the transfer path might use it. Internet Mail standards provide for a single Message-ID; however more than one is sometimes assigned.

Like a mailbox address, a Message-ID has two distinct parts, divided by an at-sign ("@"). The right-hand side is globally interpreted and specifies the ADMD or host assigning the identifier. The left-hand side contains a string that is globally opaque and serves to uniquely identify the message within the domain referenced on the right-hand side. The duration of uniqueness for the message identifier is undefined.

When a message is revised in any way, the question of whether to assign a new Message-ID requires a subjective assessment, deciding whether the editorial content has been changed enough to constitute a new message. [[RFC2822](#)] says "a message identifier pertains to exactly one instantiation of a particular message; subsequent revisions to the message each receive new message identifiers." However real-world experience dictates some flexibility. An

impossible test is whether the recipient will consider the new message to be equivalent to the old. 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.

The best that can be offered, here, are some guidelines and examples:

- o If a message is changed only in terms of form, such as character-encoding, it clearly is still the same message.
- o If a message has minor additions to the content, such as a mailing list tag at the beginning of the [RFC2822](#).Subject header field, or some mailing list administrative information added to the end of the primary body-part's text, then it probably is still the same message.
- o If a message has viruses deleted from it, it probably is still the same message.
- o If a message has offensive words deleted from it, then some recipients will consider it the same message, but some will not.
- o If a message is translated into a different language, then some recipients will consider it the same message, but some will not.

The absence of objective, precise criteria for Message-ID re-generation, along with the absence of 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. Hence it is not appropriate to use the Message-ID for any process that might be called "security".

[3.3.2.](#) ENVID

The ENVID (envelope identifier) is a tag that is primarily for use within Delivery Status Notifications (DSN), so that the Bounce Address ([RFC2821](#).MailFrom) recipient can correlate the DSN with a particular message. [[RFC3461](#)] The ENVID is therefore used from one message posting, until the directly-resulting message deliveries. It

does not survive re-postings.

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 Bounce Address.

[4.](#) Services and Standards

Internet Mail's architecture distinguishes among six different types of functional components, arranged to support a store-and-forward service architecture:

Crocker	Expires November 7, 2007	[Page 18]
---------	--------------------------	-----------

Internet-Draft	EMail Architecture	May 2007
----------------	--------------------	----------

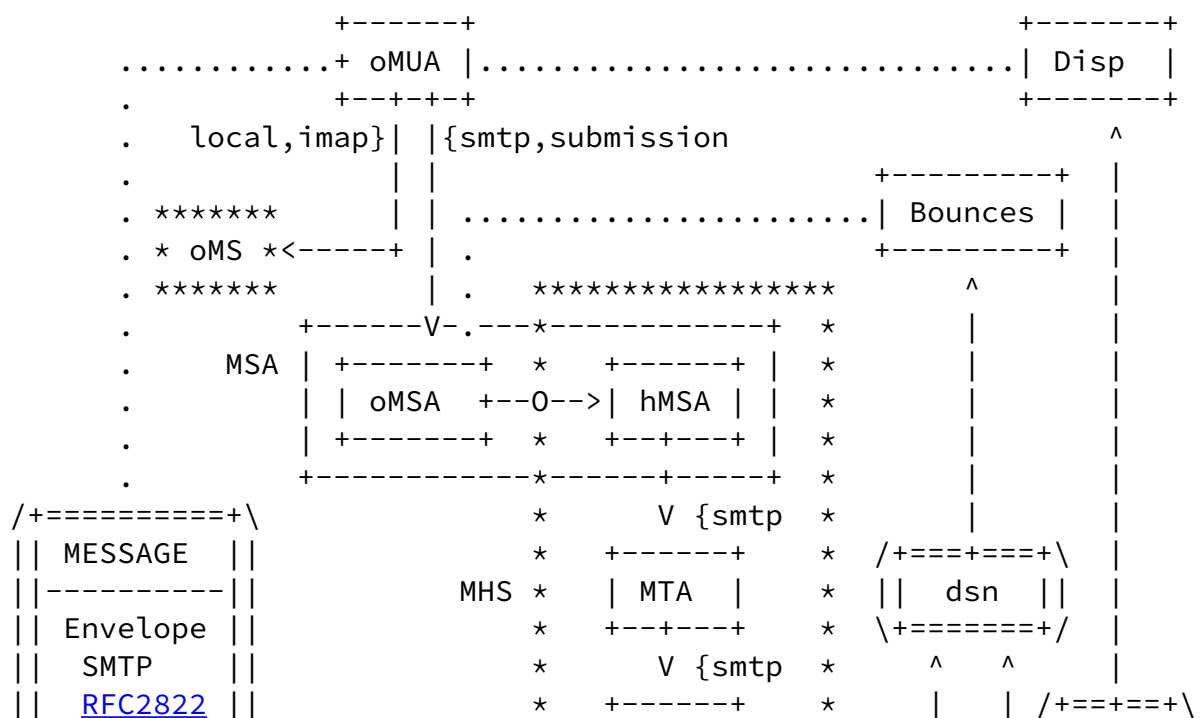
- * Message
- * Mail User Agent (MUA)
- * Message Submission Agent (MSA)
- * Message Transfer Agent (MTA)
- * Message Delivery Agent (MDA)
- * Message Store (MS)

This section describes each functional component for Internet Mail, and the standards-based protocols that are associated with their operation.

Software implementations of these architectural components often compress them, such as having the same software do MSA, MTA and MDA functions. However the requirements for each of these components of the service are becoming more extensive. So their separation is increasingly common.

NOTE:

The following figure shows function modules and the standardized protocols used between them. Additional protocols and configurations are possible. Boxes defined by asterisks (*) represent functions that often are distributed among two or more systems.



lines of text, or it may be a MIME tree of multi-media subordinate objects, called body-parts, or attachments. [[RFC2045](#)], [[RFC2046](#)], [[RFC2047](#)], [[RFC4288](#)], [[RFC4289](#)], [[RFC2049](#)].

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

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 [RFC2821](#).MailFrom address. The mailbox for this is shown as Bounces in Figure 5. It provides information about message transit, such as transmission errors or successful delivery. [[RFC3461](#)]

Message Disposition Notification (MDN):

A Message Disposition Notification (MDN) is a message that provides information about user-level, Recipient-side message 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 address(es). The mailbox for this is shown as Disp in Figure 5. It

Message Filtering (SIEVE):

SIEVE is a scripting language that permits specifying conditions for differential handling of mail, typically at the time of delivery. [[RFC3028](#)] It can be conveyed in a variety of ways, as a MIME part. Figure 5 shows a Sieve specification 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. Hence the Figure shows only one relationship, for (relative) simplicity.

[4.1.1.](#) Envelope

Information that is directly used by, or produced by, the MHS is called the "envelope". It controls and records handling activities by the transfer service. Internet Mail has a fragmented framework for handling this "handling" information. The envelope exists partly in the transfer protocol SMTP [[RFC2821](#)] and partly in the message object [[RFC2822](#)]. The SMTP specification uses the term to refer only to the transfer-protocol information.

NOTE:

Due to the frequent use of the term "envelope" to refer only to SMTP constructs, there has been some call for using a different term, to label the larger set of information defined here. So far, no alternative term has developed any community support.

Direct envelope addressing information, as well as optional transfer directives, are carried within the SMTP control channel. Other envelope information, such as trace records, is carried within the message object header fields. Upon delivery, some SMTP-level envelope information is typically encoded within additional message object header fields, such as Return-Path. [[RFC2821](#)], [[RFC2822](#)]

[4.1.2.](#) Header Fields

Header fields are attribute name/value pairs covering an extensible range of email service, user content and user transaction meta-information. The core set of header fields is defined in [[RFC2822](#)], [[RFC0822](#)]. It is common to extend this set, for different applications. Procedures for registering header fields are defined in [[RFC4021](#)]. An extensive set of existing header field registrations is provided in [[RFC3864](#)].

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

[4.1.3.](#) Body

The body of a message might simply be lines of ASCII text or it might be hierarchically structured into a composition of multi-media body-part attachments, using MIME. [[RFC2045](#)], [[RFC2046](#)], [[RFC2047](#)], [[RFC4288](#)], [[RFC2049](#)] MIME structures each body-part into a recursive set of MIME header field meta-data and MIME Content sections.

[4.1.4.](#) Identity References in a Message

For a message in transit, the core uses of identifiers combine into:

Layer	Field	Set By
Message Body	MIME Header	Originator
Message header fields	From	Originator
	Sender	Source
	Reply-To	Originator
	To, CC, BCC	Originator
	Message-ID	Source
	Received	Source, Relay, Dest
	Return-Path	MDA, from MailFrom
	Resent-*	Mediator
SMTP	HELO	Latest Relay Client
	MailFrom	Source
	RcptTo	Originator
IP	IP Address	Latest Relay Client

Layered Identities

The most common address-related fields are:

[RFC2822](#).From

Set by: Originator

Names and addresses for author(s) of the message content are listed in the From field.

[RFC2822](#).Reply-To

Set by: Originator

If a message Recipient sends a reply message that would otherwise use the [RFC2822](#).From field address(es) that are contained in the original message, then they are instead to use the address(es) in the [RFC2822](#).Reply-To field. In other words this field is a direct override of the From field, for responses from Recipients.

[RFC2822](#).Sender

Internet-Draft

EMail Architecture

May 2007

Set by: Source

This specifies the address responsible for submitting the message into the transfer service. For efficiency this field can be omitted if it contains the same address as [RFC2822.From](#). However this does not mean there is no Sender specified. Rather it means that that header field is virtual and that the address in the From field MUST be used.

Specification of the error return addresses -- the "Bounce" address, contained in [RFC2821.MailFrom](#) -- is made by the [RFC2822.Sender](#). Typically the Bounce address is the same as the Sender address. However some usage scenarios require it to be different.

[RFC2822.To](#), [RFC2822.CC](#)

Set by: Originator

These specify MUA Recipient addresses. However some or all of the addresses in these fields might not be present in the [RFC2821.RcptTo](#) commands, due to handling process that might transfer from the former to the latter.

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 only for their information.

[RFC2822.BCC](#)

Set by: Originator

A message might be copied to an addressee whose participation is not to be disclosed to the [RFC2822.To](#) or [RFC2822.CC](#) Recipients and, usually, not to the other BCC Recipients. The BCC header field indicates a message copy to such a Recipient.

Typically, the field lists no addresses or only lists the address of the Recipient receiving this copy. An MUA will typically make separate postings for TO and CC Recipients, versus BCC Recipients. The former will see no indication that any BCCs were sent, whereas

the latter have a BCC field present. It might be empty, contain a comment, or contain one or more BCC addresses, depending upon the preferences of the Originator.

[RFC2821](#).HELO/.EHLO

Set by: Source

The MSA can specify its hosting domain identity for the SMTP HELO or EHLO command operation.

[RFC2821](#).MailFrom

Set by: Source

This is an end-to-end string that specifies an email address for receiving return control information, such as "bounces". The name of this field is misleading, because it is not required to specify either the author or the agent responsible for submitting the message. Rather, the agent responsible for submission specifies the [RFC2821](#).MailFrom address. Ultimately the simple basis for deciding what address needs to be in the [RFC2821](#).MailFrom is to determine what address needs to be informed about transmission-level problems (and, possibly, successes.)

[RFC2821](#).RcptTo

Set by: Originator

This specifies the MUA mailbox address of a recipient. The string might not be visible in the message content header. For example, the message destination address header fields, such as [RFC2822](#).To, might specify a mailing list mailbox, while the [RFC2821](#).RcptTo address specifies a member of that list.

[RFC2821](#).Received

Set by: Source, Relay, Mediator, Dest

This indicates trace information, including originating host, relays, Mediators, and MSA host domain names and/or IP Addresses.

[RFC2821](#).Return-Path

Set by: Source

The MDA records the [RFC2821](#).MailFrom address into the [RFC2822](#).Return-Path field.

Crocker

Expires November 7, 2007

[Page 25]

Internet-Draft

EMail Architecture

May 2007

[RFC2919](#).List-Id

Set by: Mediator Originator

This 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 and the result rarely is a true domain name, listed in the domain name service -- although it can be.

[RFC2369](#).List-*

Set by: Mediator Originator

[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](#)]

[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 architecture, which is above the Internet Transport layer. Because the motivation for email, and much of its use, is for interaction

among humans, the nature and details of these protocol exchanges often are determined by the needs of human and group communication. In terms of efforts to specify behaviors, one effect of this is to require subjective guidelines, rather than strict rules, for some aspects of system behavior. Mailing Lists provide particularly salient examples of this.

4.2.1. Mail User Agent (MUA)

A Mail User Agent (MUA) works on behalf of end-users and end-user applications. It is their "representative" within the email service.

The Origination-side MUA (oMUA) 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 archival in its Message Store (oMS). An MUA's oMS will typically include 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 transmission (Sent).

Crocker

Expires November 7, 2007

[Page 26]

Internet-Draft

EMail Architecture

May 2007

The Recipient-side MUA (rMUA) works on behalf of the end-user Recipient to process received mail. This includes generating user-level return 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 can, itself, have a distributed implementation, such as a "thin" user interface module on a limited end-user device, with the bulk of the MUA functionality operated 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. A standardized approach for such scenarios is defined by [[RFC4550](#)].

A Mediator is special class of MUA. It performs message re-posting, as discussed in [Section 2.1](#).

Identity fields relevant to a typical end-user MUA include:

[RFC2822](#).From

[RFC2822](#).Reply-To

[RFC2822](#).Sender

[RFC2822](#).To, [RFC2822](#).CC

[RFC2822](#).BCC

[4.2.2](#). Message Store (MS)

An MUA can employ a long-term Message Store (MS). Figure 5 depicts an Origination-side MS (oMS) and a Recipient-side MS (rMS). There is a rich set of choices for configuring a store, because any MS may comprise a distributed set of component stores. In Figure 5, the rMS demonstrates this by showing an rMS that is located on a remote server (srMS) and an rMS that is on the same machine as the MUA (urMS). The relationship between two message stores, themselves, can vary.

As discussed in [[RFC1733](#)] the operational relationship among MSs can be --

Online:

Only a remote MS is used, with messages being accessible only when the MUA is attached to the MS, and the MUA repeatedly fetches 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 there is a connection between them. While they are disconnected, mail can continue to arrive at the rMS and the user may continue to make changes to the uMS. Upon reconnection, the two stores are re-synchronized.

[4.3.](#) MHS-Level Services

[4.3.1.](#) Mail Submission Agent (MSA)

A Mail Submission Agent (MSA) accepts the message submission from the oMUA and enforces the policies of the hosting ADMD and the requirements of Internet standards. An MSA represents an unusual functional dichotomy. A portion of its task is to represent MUA (uMSA) interests during message posting, to facilitate posting success, and another portion is to represent MHS (hMSA) interests. This is best modeled, as shown in Figure 5, with two sub-components, one for the oMUA (oMSA) and one for the MHS (hMSA)

The hMSA's function is to take 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 oMSA's is to perform final message preparation for submission and to effect the transfer of responsibility to the MHS, via the hMSA. The amount of preparation will depend upon the local implementations. Examples of oMSA tasks could be to add header fields, such as Date: and Message-ID, to modify portions of the message from local notations to Internet standards, such as expanding an address to its formal [RFC2822](#) representation.

Historically, standards-based MUA/MSA interactions have used SMTP [[RFC2821](#)]. A recent alternative is SUBMISSION [[RFC4409](#)]. Although SUBMISSION derives from SMTP, it uses a separate TCP port and imposes distinct requirements, such as access authorization.

Identities relevant to the MSA include:

[RFC2821](#).HELO/.EHLO

[RFC2821](#).MailFrom

[RFC2821](#).RcptTo

[RFC2821](#).Received

[4.3.2](#). Mail Transfer Agent (MTA)

A Mail 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 Recipient(s). 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 make changes to addresses in the envelope or reformulate the editorial content. Hence a change in data form, such as to the MIME Content-Transfer-Encoding, is within the purview of an MTA, whereas removal or replacement of body content is not. Also it can add trace information. 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.

Internet Mail primarily uses SMTP [[RFC2821](#)], [[RFC0821](#)] to effect point-to-point transfers between peer MTAs. Other transfer mechanisms include Batch SMTP [[RFC2442](#)] and ODMR [[RFC2645](#)]. As with most network layer mechanisms, Internet Mail's SMTP supports a basic level of reliability, by virtue of providing for retransmission after a temporary transfer failure. Contrary to typical packet switches (and Instant Messaging services) Internet Mail MTAs typically store messages in a manner that allows recovery across service interruptions, such as host system shutdown. However the degree of such robustness and persistence by an MTA can be highly variable.

The primary "routing" mechanism for Internet Mail is the DNS MX record [[RFC1035](#)], which specifies a host through which the queried domain can be reached. This presumes a public -- or at least a common -- backbone that permits any attached host to connect to any other.

Identities relevant to the MTA include:

[RFC2821](#).HELO/.EHLO

[RFC2821](#).MailFrom

[RFC2821](#).RcptTo

[RFC2822](#).Received

Set by: Relay Server

[4.3.3](#). Mail Delivery Agent (MDA)

A Mail Delivery Agent (MDA) delivers email to the Recipient's mailbox. It can provide distinctive, address-based functionality, made possible by its detailed knowledge of the properties of the destination address. This knowledge 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 must know where to deliver the message.

As with 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. The MHS portion (hMDA) primarily functions as a server SMTP engine. A common additional role is to re-direct 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 are desired by the recipient.

Using Internet protocols, delivery can be effected by a variety of standard protocols. When coupled with an internal local mechanism, SMTP [[RFC2821](#)] and LMTP [[RFC2033](#)] permit "push" delivery to the Recipient system, at the initiative of the upstream email service. POP [[RFC1939](#)] and IMAP [[RFC3501](#)] are used for "pull" delivery at the initiative of the Recipient system. POP and IMAP can also be used for repeated access to messages on a remote MS.

Identities relevant to the MDA include:

[RFC2821](#).Return-Path

Internet-Draft

EMail Architecture

May 2007

Set by: Originator Source or Mediator Source

The MDA records the [RFC2821](#).MailFrom address into the [RFC2822](#).Return-Path field.

[RFC2822](#).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.

[5](#). Mediators

Basic email transfer from an Originator to the specified 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 User-level sequence of postings and deliveries, through Mediators. A Mediator forwards a message, through a re-posting process. The Mediator does share some functionality with basic MTA relaying, but it enjoys a degree of freedom with both addressing and content that is not available to MTAs.

[RFC2821](#).HELO/.EHLO

Set by: Mediator Source

[RFC2821](#).MailFrom

Set by: Originator Source or Mediator Source

[RFC2821](#).RcptTo

Set by: Mediator Originator

[RFC2821](#).Received

Set by: Mediator Dest

The salient aspect of a Mediator, that distinguishes it from any other MUA creating an entirely new 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 message creation that are NOT performed by Mediators include --

New message that forwards an existing message:

This action rather curiously provides a basic template for a class of Mediators. However for its typical occurrence it is not itself an example of a Mediator. The new message is viewed as being from the Agent doing the forwarding, rather than being from the original Originator.

A new message encapsulates the original message and is seen as strictly "from" the Mediator. The Mediator might add commentary and certainly has the opportunity to modify the original message content. The forwarded message is therefore independent of the original message exchange and creates a new message dialogue. However the final Recipient sees the contained message as from the original Originator.

Reply:

When a Recipient formulates a response back to the original message's author, the new message is not typically viewed as being a "forwarding" of the original. Its focus is the new content, although it might contain all or part of the material in the original message. Therefore the earlier material is merely contextual and secondary.

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 tone of the new message is that it is primarily commentary from a new Originator,

similar to a Reply.

The remainder of this section describes common examples of Mediators.

[5.1.](#) Aliasing

Aliasing is a simple re-addressing facility that is available in most MDA implementations. It is performed just before placing a message into the specified Recipient's mailbox. Instead the message is submitted back to the transfer service, for delivery to one or more alternate addresses. Although typically implemented as part of an MDA, this facility is strictly a Recipient user function. It

Crocker

Expires November 7, 2007

[Page 32]

Internet-Draft

EMail Architecture

May 2007

resubmits the message, replacing the envelope address, on behalf of the mailbox address that was listed in the envelope.

What is most distinctive about this forwarding mechanism is how closely it compares to normal MTA store-and-forward Relaying. Its only interesting difference is that it changes the [RFC2821](#).RcptTo value. Having the change be this small makes it easy to view aliasing as a part of the lower-level mail relaying activity. However the small change has a large semantic impact: The designated recipient has chosen a new recipient. Hence that original recipient SHOULD become responsible for any handling issues. This change would be reflected by replacing the message's [RFC2821](#).MailFrom address to be one within the scope of the ADMD doing the aliasing.

An MDA that is re-posting a message to an alias typically changes only envelope information:

[RFC2822](#).TO, [RFC2822](#).CC, [RFC2822](#).BCC

Set by: Originator

These retain their original addresses.

[RFC2821](#).RcptTo

Set by: Mediator Originator

This field contains an alias address.

[RFC2821](#).MailFrom

Set by: Originator Source or Mediator Source

The agent responsible for submission to an alias address will often retain the original address to receive handling Bounces. The benefit of retaining the original MailFrom value is to ensure that the origination-side agent knows that there has been a delivery problem. On the other hand, the responsibility for the problem usually lies with the Recipient, since the Alias mechanism is strictly under the Recipient's control.

[RFC2821](#).Received

Crocker

Expires November 7, 2007

[Page 33]

Internet-Draft

EMail Architecture

May 2007

Set by: Mediator Dest

The agent 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 therefore include everything from original posting through final delivery to a final delivery.

[5.2](#). Re-Sending

Also called Re-Directing, Re-Sending differs from Forwarding by virtue of having the Mediator "splice" a message's addressing information, to connect the Originator of the original message and the Recipient of the new message. This permits them to have direct exchange, using their normal MUA Reply functions. Hence the new Recipient sees the message as being From the original Originator, even if the Mediator adds commentary.

Identities specified in a resent message include

[RFC2822](#).From

Set by: original Originator

Names and email addresses for the original author(s) of the message content are retained. The free-form (display-name) portion of the address might be modified to provide informal reference to the agent responsible for the redirection.

[RFC2822](#).Reply-To

Set by: original Originator

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

[RFC2822](#).Sender

Set by: Originator Source or Mediator Source

[RFC2822](#).TO, [RFC2822](#).CC, [RFC2822](#).BCC

Set by: original Originator

These specify the original message Recipients.

[RFC2822](#).Resent-From

Set by: Mediator Originator

The address of the original Recipient who is redirecting the message. Otherwise the same rules apply for the Resent-From field as for an original [RFC2822](#).From field

[RFC2822](#).Resent-Sender

Set by: Mediator Source

The address of the agent responsible for re-submitting the

message. As with [RFC2822](#).Sender, this field is often omitted when it would merely contain the same address as [RFC2822](#).Resent-From.

[RFC2822](#).Resent-To, [RFC2822](#).Resent-cc, [RFC2822](#).Resent-bcc:

Set by: Mediator Originator

The addresses of the new Recipients who will now be able to reply to the original author.

[RFC2821](#).MailFrom

Set by: Mediator Source

The agent responsible for re-submission ([RFC2822](#).Resent-Sender) is also responsible for specifying the new MailFrom address.

[RFC2821](#).RcptTo

Set by: Mediator Originator

This will contain the address of a new Recipient

[RFC2822](#).Received

Set by: Mediator Dest

When resending a message the submission agent can record a Received header field, to indicate the transition from original posting to resubmission.

[5.3](#). Mailing Lists

Mailing lists have explicit email addresses and they re-post messages to a list of subscribed members. The Mailing List Actor performs a task that can be viewed as an elaboration of the Re-Director role. In addition to sending the new message to a potentially large number of new Recipients, the Mediator can modify content, such as deleting attachments, formatting conversion, and adding list-specific

comments. In addition, archiving list messages is common. Still the message retains characteristics of being "from" the original Originator.

Identities relevant to a mailing list processor, when submitting a message, include:

[RFC2919](#).List-Id

Set by: Mediator Originator

[RFC2369](#).List-*

Set by: Mediator Originator

[RFC2822](#).From

Set by: original Originator

Names and email addresses for the original author(s) of the message content are specified -- or, rather, retained.

[RFC2822](#).Reply-To

Set by: original Originator or Mediator Originator

[RFC2822](#).Sender

Set by: Originator Source or Mediator Source

This will usually specify the address of the agent responsible for mailing list operations. However some mailing lists operate in a manner very similar to a simple MTA Relay, so that they preserve as much of the original handling information as possible, including the original [RFC2822](#).Sender field.

Set by: original Originator

These usually contains the original list of Recipient addresses.

[RFC2821](#).MailFrom

Set by: Originator Source or Mediator Source

This can contain the original address to be notified of transmission issues, or the mailing list agent can set it to contain a new Notification address. Typically the value is set to a new address, so that mailing list members and posters are not burdened with transmission-related Bounces.

[RFC2821](#).RcptTo

Set by: Mediator Originator

This contains the address of a mailing list member.

[RFC2821](#).Received

Set by: Mediator Dest

A Mailing List Agent can record a Received header field, to indicate the transition from original posting to mailing list forwarding. The Agent can choose to have the message retain the original set of Received header fields or can choose to remove them. In the latter case it can ensure that the original Received header fields are otherwise available, to ensure later accountability and diagnostic access to them.

[5.4](#). Gateways

A Gateway performs the basic routing and transfer work of message relaying, but it also may make any message or address modifications that are needed to send the message into a messaging environment that operates according to 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 have technical similarity, but can have significant administrative differences, it is easy to think that a Gateway is merely an MTA.

The critical distinction between an MTA and a Gateway is that the

latter transforms addresses and/or message content, in order to map between the standards of two, different messaging services. In virtually all cases, this mapping process results in some degree of semantic loss. The challenge of Gateway design is to minimize this loss.

A Gateway can set any identity field available to a regular MUA. Identities typically relevant to Gateways include:

[RFC2822](#).From

Set by: original Originator

Names and email addresses for the original author(s) of the message content are retained. As for all original addressing information in the message, the Gateway can translate addresses in whatever way will allow them continue to be useful in the target environment.

[RFC2822](#).Reply-To

Set by: original Originator

The Gateway SHOULD retain this information, if it is originally present. The ability to perform a successful reply by a Gatewayed Recipient is a typical test of Gateway functionality.

[RFC2822](#).Sender

Set by: Originator Source or Mediator Source

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

[RFC2822](#).TO, [RFC2822](#).CC, [RFC2822](#).BCC

Set by: original Recipient

These usually retain their original addresses.

[RFC2821](#).MailFrom

Set by: Originator Source or Mediator Source

The agent responsible for gatewaying the message can choose to specify a new address to receive handling notices.

[RFC2822](#).Received

Set by: Mediator Dest

The Gateway can record a Received header field, to indicate the transition from original posting to the new messaging environment.

[5.5.](#) Boundary Filter

Organizations often enforce security boundaries by subjecting messages to analysis, for conformance with the organization's 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 will result in the addition of content that records the actions.

[6.](#) Security Considerations

This document does not specify any new Internet Mail functionality. Consequently it is not intended to introduce any security considerations.

However its discussion of the roles and responsibilities for different mail service modules, and the information they create, highlights the considerable degree to which security issues are present when implementing any component of the Internet Mail service. In addition, email transfer protocols can operate over authenticated and/or encrypted links, and message content or authorship can be authenticated or encrypted.

[7.](#) IANA Considerations

This document has no actions for IANA.

[8.](#) References

[8.1.](#) Normative

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

Crocker

Expires November 7, 2007

[Page 39]

Internet-Draft

EMail Architecture

May 2007

- [RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, [RFC 822](#), August 1982.
- [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.
- [RFC2033] Myers, J., "Local Mail Transfer Protocol", [RFC 2033](#), October 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.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

- [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.
- [RFC2442] "The Batch SMTP Media Type", [RFC 2442](#), November 1998.
- [RFC2645] "On-Demand Mail Relay (ODMR) SMTP with Dynamic IP Addresses", [RFC 2645](#), August 1999.
- [RFC2821] Klensin, J., "Simple Mail Transfer Protocol", [RFC 2821](#), April 2001.

Crocker

Expires November 7, 2007

[Page 40]

Internet-Draft

EMail Architecture

May 2007

- [RFC2822] Resnick, P., "Internet Message Format", [RFC 2822](#), April 2001.
- [RFC2919] Chandhok, R. and G. Wenger, "List-Id: A Structured Field and Namespace for the Identification of Mailing Lists", [RFC 2919](#), March 2001.
- [RFC3028] Showalter, T., "Sieve: A Mail Filtering Language", [RFC 3028](#), January 2001.
- [RFC3192] Allocchio, C., "Minimal FAX address format in Internet Mail", [RFC 2304](#), 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.
- [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.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", [RFC 3864](#), September 2004.
- [RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", [RFC 4021](#), March 2005.
- [RFC4288] Freed, N., Klensin, J., and J. Postel, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [RFC4289] Freed, N., Klensin, J., and J. Postel, "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.

Crocker

Expires November 7, 2007

[Page 41]

Internet-Draft

E-Mail Architecture

May 2007

- [RFC4550] Maes, S., , S., and Isode Ltd., "Internet Email to Support Diverse Service Environments (Lemonade) Profile", June 2006.

[8.2.](#) Descriptive

[ID-spamops]

Hutzler, C., Crocker, D., Resnick, P., Sanderson, R., and E. Allman, "Email Submission Between Independent Networks", [draft-spamops-00](#) (work in progress), March 2004.

- [RFC1733] Crispin, M., "Distributed Electronic Models in IMAP4", December 1994.

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

- [RFC3801] Vaudreuil, G. and G. Parsons, "", [RFC 3801](#), June 2004.
- [RFC4142] Crocker, D. and G. Klyne, "Full-mode Fax Profile for Internet Mail: FFPIM", December 2005.
- [Tussle] Clark, D., Wroclawski, J., Sollins, K., and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet", ACM SIGCOMM, 2002.

[Appendix A](#). Acknowledgements

This work derives from a section in [draft-hutzler-spamops](#).
[ID-spamops] Discussion of the Source 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.

Later reviews and suggestions were provided by Nathaniel Borenstein, Ed Bradford, Cyrus Daboo, Frank Ellermann, Tony Finch, Ned Freed, Eric Hall, Brad Knowles, John Leslie, Bruce Lilly, Mark E. Mallett, David MacQuigg, Chris Newman, Daryl Odnert, Rahmat M. Samik-Ibrahim, Marshall Rose, Hector Santos, Jochen Topf, Willemien Hoogendoorn, Valdis Kletnieks.

Diligent proof-reading was performed by Bruce Lilly.

Author's Address

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

Phone: +1.408.246.8253
Email: dcrocker@bbiw.net

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

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

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

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

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).