

SMTP  
Internet-Draft  
Expires: July 27, 2005

D. Crocker  
Brandenburg InternetWorking  
January 26, 2005

Internet Mail Architecture  
draft-crocker-email-arch-02

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [section 3 of RFC 3667](#). 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 become aware will be disclosed, in accordance with [RFC 3668](#).

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 July 27, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Over its thirty year history, Internet mail has undergone significant changes in scale and complexity. The first standardized architecture for email specified a simple split between the user world and the transmission world, in the form of Mail User Agents (MUA) and Mail Transfer Agents (MTA). Over time each of these has divided into multiple, specialized modules. Public discussion and agreement about the nature of the changes to Internet mail has not kept pace, and abuses of the Internet mail service have brought these issues into

Internet-Draft

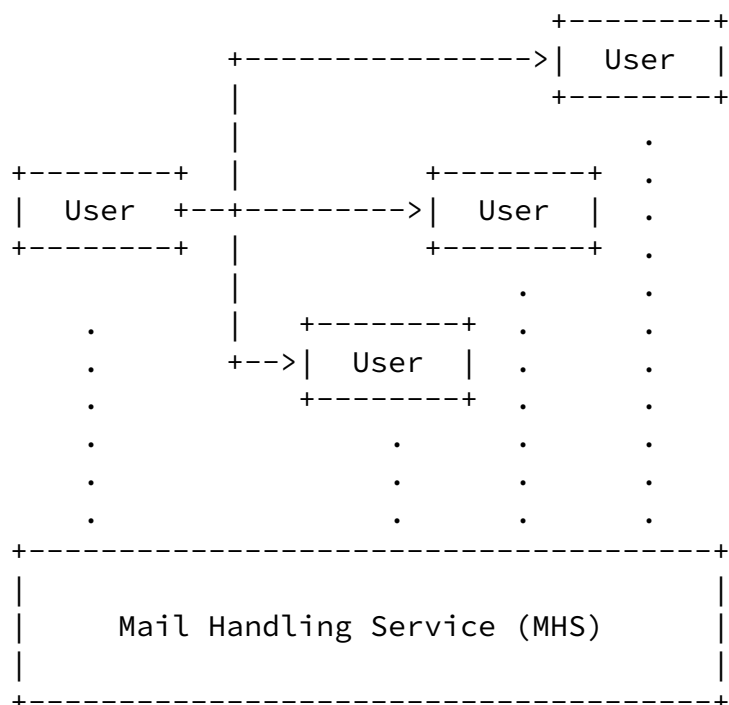
EMail Architecture

January 2005

stark relief. This draft offers clarifications and enhancements, to provide a more consistent base for community discussion of email service problems and proposed email service enhancements.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1</a>	Service Overview . . . . .	<a href="#">4</a>
<a href="#">1.2</a>	Document Changes . . . . .	<a href="#">5</a>
<a href="#">1.3</a>	Discussion venue . . . . .	<a href="#">5</a>
<a href="#">2.</a>	Email Actor Roles . . . . .	<a href="#">5</a>
<a href="#">2.1</a>	User-Level Actors . . . . .	<a href="#">6</a>
<a href="#">2.2</a>	Transfer-Level Actors . . . . .	<a href="#">8</a>
<a href="#">2.3</a>	Administrative Actors . . . . .	<a href="#">11</a>
<a href="#">3.</a>	Email Identities . . . . .	<a href="#">11</a>
<a href="#">3.1</a>	Mailbox Addresses . . . . .	<a href="#">12</a>
<a href="#">3.2</a>	Domain Names . . . . .	<a href="#">13</a>
<a href="#">3.3</a>	Message Identifiers . . . . .	<a href="#">13</a>
<a href="#">3.4</a>	Identity Referencing Convention . . . . .	<a href="#">13</a>
<a href="#">4.</a>	Protocols and Services . . . . .	<a href="#">13</a>
<a href="#">4.1</a>	Service Components . . . . .	<a href="#">15</a>
<a href="#">4.2</a>	Operational Configuration . . . . .	<a href="#">21</a>
<a href="#">4.3</a>	Layers of Identity References . . . . .	<a href="#">21</a>
<a href="#">5.</a>	Message Data . . . . .	<a href="#">22</a>
<a href="#">5.1</a>	Envelope . . . . .	<a href="#">22</a>
<a href="#">5.2</a>	Message Header Fields . . . . .	<a href="#">22</a>
<a href="#">5.3</a>	Body . . . . .	<a href="#">23</a>
<a href="#">6.</a>	Two Levels of Store-And-Forward . . . . .	<a href="#">23</a>
<a href="#">6.1</a>	MTA Relaying . . . . .	<a href="#">23</a>
<a href="#">6.2</a>	MUA Forwarding . . . . .	<a href="#">23</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">30</a>
<a href="#">8.</a>	References . . . . .	<a href="#">31</a>
	Author's Address . . . . .	<a href="#">33</a>
<a href="#">A.</a>	Acknowledgements . . . . .	<a href="#">33</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">34</a>



## Figure 1: Basic Email Service Model

Public discussion and agreement about terms of reference have not kept pace with the changes, and abuses of the Internet mail service have brought this into stark relief. So, it is necessary to produce a revised architecture. However it is important that the original distinction between user-level concerns and transfer-level concerns be retained. This becomes challenging when the user-level exchange is, itself, a sequence, such as with group dialogue or organizational message flow, as occurs with a purchase approval process. It is easy to confuse this user-level activity with the underlying mail transmission service exchanges.

Crocker

Expires July 27, 2005

[Page 3]

---

Internet-Draft

EMail Architecture

January 2005

For Internet mail, the term "end-to-end" usually refers to single posting and the set of deliveries resulting from a single transiting of the MHS. However, note that specialized 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 and EDI over email.

The current draft seeks to:

1. Document changes that have taken place in refining the email model
2. Clarify functional roles for the architectural components
3. Clarify identity-related issues, across the email service
4. Provide a common venue for further defining and citing modern Internet mail architecture

### [1.1](#) Service Overview

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

1. An email object

2. Global addressing
3. A connected sequence of point-to-point transfer mechanisms
4. No prior arrangement between originator and recipient
5. No prior arrangement between point-to-point transfer services, over the open Internet

The end-to-end portion of the service is the message. Broadly the message, itself, is divided between handling control information and user message payload.

A precept to the design of Internet mail is to permit user-to-user and MTA-to-MTA interoperability with no prior, direct administrative arrangement. That is, all participants rely on having the core services be universally supported, either directly or through gateways that translate between Internet mail standards and other email conventions.

For localized environments (edge networks) prior, administrative arrangement 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, the server performs explicit validation of the client's identity.

## [1.2](#) Document Changes

The major changes from the previous version of this document are:

Overall: Clarify roles and responsibilities

Diagrams: Revised diagrams and tightened things up

Distinct architectural 'sections': Added concept of ADMDs, as operational layer, separate from functional or architectural layer. Added user "layer", as distinct from transfer. Introduced 'mediator'.

### [1.3](#) Discussion venue

NOTE: This document is the work of a single person, about a topic with considerable diversity of views. It is certain to be incomplete and inaccurate. Some errors simply need to be reported; they will get fixed. Others need to be discussed by the community, because the real requirement is to develop common community views. To this end, please treat the draft as a touchstone for public discussion.

Discussion about this document should be directed to the: <mailto:ietf-smtp@imc.org> mailing list. The IETF-SMTP mailing list <<http://www.imc.org/ietf-smtp/index.htm>> is the most active, long-standing venue for discussing email architecture. Although this list is primarily for discussing only the SMTP protocol, it is recommended that discussion of this draft take place on that mailing list. This list tends to attend to end-to-end infrastructure and architecture issues more than other email-related mailing lists.

## [2.](#) Email Actor Roles

Discussion of email architecture requires distinguishing different actors within the service, and being clear about the job each performs. The best way to maintain the distinction between user activity and handling activities is to depict their details in separate diagrams. Current Internet mail provides only a small set of capabilities for supporting different kinds of ongoing, user-level

exchanges.

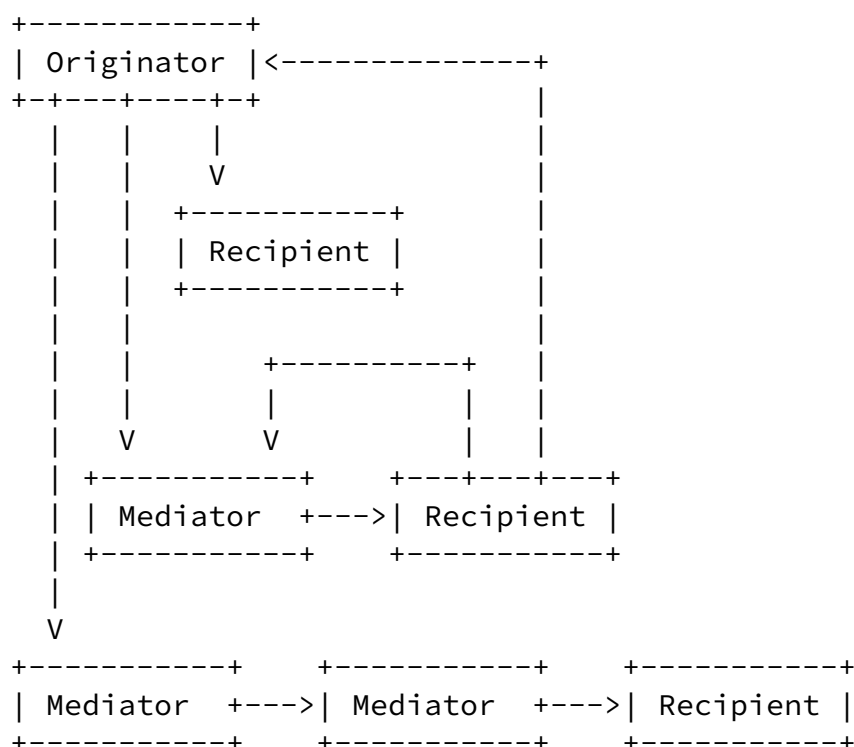
Although related to a technical architecture, the focus of a discussions on Actors is on participant responsibilities, rather than functional modules. Hence the labels used are different than for classic email architecture diagrams. The figures depict the relationships among the Actors. Actors often will be associated with entirely independent organizations from other Actors who are participating in the email service.

### [2.1](#) User-Level Actors

Users are the sources and sinks of messages. They may have an exchange that iterates and they may expand or contract the set of

users participating in a set of exchanges.

In Internet Mail there are three, basic types of user-level Actors: Originators, Recipients, and Mediators. From the User-level perspective all mail transfer activities are performed by a monolithic, shared handling service. Users are customers of this service. The following depicts the relationships among them.



The functions of these Actors are:

### 2.1.1 Originator

Also called "Author", this is the user-level participant responsible

for creating original content and requesting its transmission. The Mail Handling Service operates to send and deliver mail among Originators and Recipients.

### 2.1.2 Recipient

The Recipient is a consumer of delivered content.

A recipient may close the user-level communication loop by creating and submitting a new message that replies to an originator. An automated, or semi-automated form of reply informs the Originator about the Recipient's disposition of the message.

### [2.1.3](#) Mediator

A Mediator receives, aggregates, reformulates and distributes messages as part of a potentially-protracted, higher-level exchange among users. A Mediator is viewed by the Mail Handling Service, when the Mediator's address is specified in the envelope. When submitting messages, the Mediator is an Originator. What is distinctive is that a Mediator preserves Originator information of the message(s) it reformulates, but makes meaningful changes to the content. Hence the Mail Handling Service sees a new message, but Users receive a message that is interpreted as primarily being from the author of the original message. The role of a Mediator permits distinct, active creativity, rather than being limited the more passive job of merely connecting together other participants. Hence it is really the Mediator that is responsible for the new message.

A Mediator's task may be complex, contingent and creative, such as by modifying and adding content or regulating which users may 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 might also receive replies. That is, a Mediator is a full-fledged User.

Specialized Mediators include:

Forwarder: A new message encapsulates the original message and is seen as strictly "from" the Mediator. However the Mediator might add commentary and certainly has the opportunity to modify the original message content.



Redirector: Redirection differs from Forwarding by virtue of having the Mediator "splice" communication between the Originator of the original message and the Recipient of the new message. Hence the new Recipient sees the message as being From the original Originator.

Mailing List: This Actor performs a task that can be viewed as an elaboration of the Redirector role. In addition to sending the new message to a potentially large number of new Recipients, content might be modified, such as deletion of attachments, formatting conversion, and addition of list-specific comments. In addition, archival of list messages is common.

Annotator: The integrity of the original message is preserved, but one or more comments about the message are added in a manner that distinguishes commentary from original text.

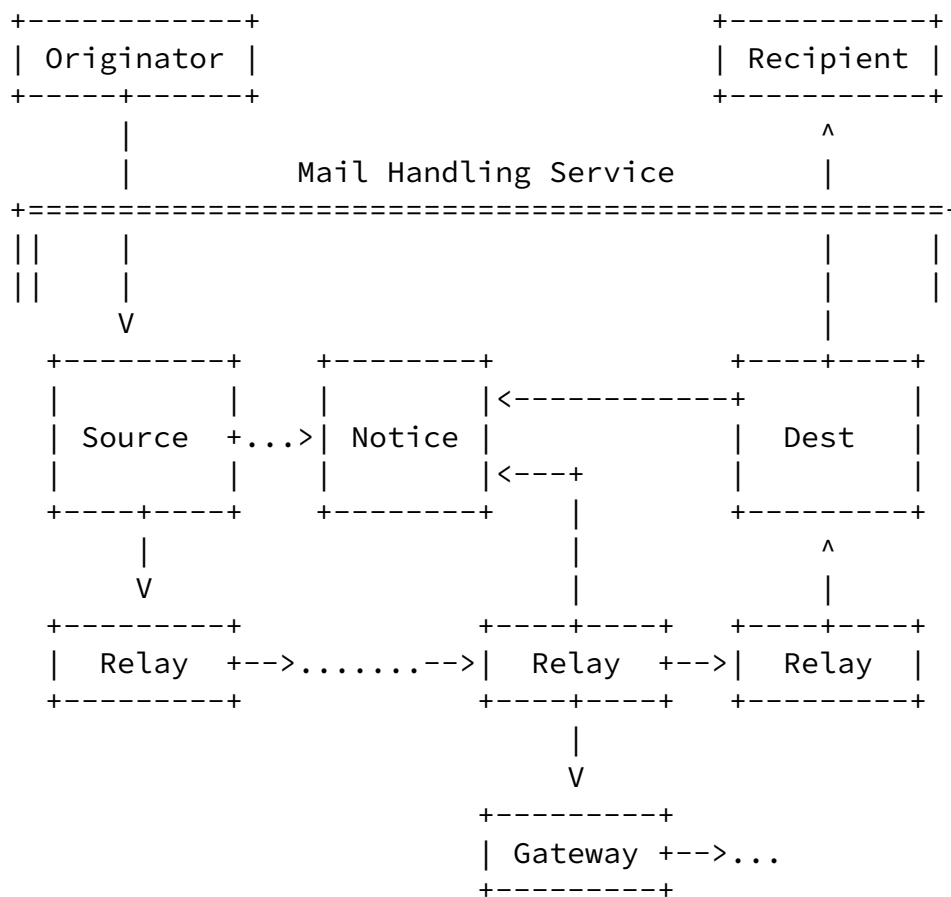
Adaptor: {per Ned Freed}

Security Filter: Organizations often enforce security boundaries by having message subjected to analysis for conformance with the organization's safety policies. Examples are detection of content classed as spam or a virus. A Security 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.

## [2.2](#) Transfer-Level Actors

The Mail Handling Service 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 envelope. 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 service.

The following depicts the relationships among transfer participants in Internet Mail. It shows Source as distinct from the Originator, although it is common for them to be the same actor. The figure also shows multiple Relays in the sequence. It is legal to have only one, and for intra-organization mail services, this is common.



### [2.2.1](#) Source

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

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 represent the local operator of the Mail Handling Service.

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](#) Notices Handler

Transfer efforts might result in the generation of service reporting information about failures or completions. These Transfer or Delivery notification messages are sent to an address that is specified by the Source. A Notices handling address (also known as Bounce or Return address) might have no characteristics in common the with address of the Originator or Source.

### [2.2.3](#) Relay

A mail relay performs email transfer-service routing and store-and-forward. It adds envelope-related handling information and then (re-)transmits the message on towards its recipient(s). A Relay does not modify the message contents.

A basic transfer operation is between a client and a server Relay. A set of Relays composes a Mail Handling Service network. This is above any underlying packet-switching network that they might be using.

Aborting message transfer results in having the Relay become an Originator and send an error message to the Notifications (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 special form of Relay that interconnects heterogeneous mail services. Differences between the services can be as small as minor syntax variations, but usually encompass much more basic, semantic distinctions. For example, the concept of an email address

might be as different as a hierarchical, machine-specific address versus a flat, global name space. Or between text-only and multi-media. Hence, the Relay function of a gateway is the minor component. The significant challenge is in 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 can send a message to a recipient, without requiring any changes to the components in the originator's mail service or the recipient's mail service, 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 formulate a Reply?

### [2.3](#) Administrative Actors

Operation of Internet mail services is apportioned to different providers (or operators) each is composed of an independent Administrative Domain. 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 Administrative Domain potentially having a complex arrangement of functional components.

The interactions between functional components within an Administrative Domain are subject to the policies of that domain. Policies can cover such things as reliability, access control, accountability and content evaluation and may be implemented in different functional components, according to the needs of the Administrative Domain.

#### [2.3.1](#) Provider

Providers operate component services or sets of services. It is possible for Providers to host services for other Providers. Common examples are:

Enterprise Service Providers: Operating an organization's internal data and/or mail operations.

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

Mail Service Providers: Operate 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 insure the health of the overall Internet Mail Service.

### 3. Email Identities

Internet mail uses three forms of identity. The most common is the mailbox address <addr-spec> [[RFC2822](#)]. The other two forms are the

Crocker

Expires July 27, 2005

[Page 11]

---

Internet-Draft

E-Mail Architecture

January 2005

<domain name> [[RFC1034](#)] and message identifier [[RFC2822](#)].

#### 3.1 Mailbox Addresses

An addr-spec has two distinct parts, divided by an at-sign ("@"). The right-hand side contains a globally interpreted name for an administrative domain. This domain name might refer to an entire organization, or to a collection of machines integrated into a homogeneous service, or to a single machine. Domain names are defined and operated through the DNS [[RFC1034](#)], [[RFC1035](#)].

The left-hand side 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 in the address's right-hand side. All other entities must treat the local-part as a uninterpreted, literal string and must preserve all of its original details. As such, its distribution is equivalent to sending a "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 by the same participant. However it must be stressed 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 add-spec.

A small class of addresses 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 [[RFC2421](#)], such as "+16137637582@vpim.example.com", and iFax [[RFC2304](#)], 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 name usually maps to one or more IP Addresses. A domain name can be administered to refer to individual users, but this is not common practice. The name is structure as a hierarchical sequence of sub-names, separated by dots (".").

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

### [3.3](#) Message Identifiers

Message identifiers have two distinct parts, divided by an at-sign ("@"). The right-hand side contains a globally interpreted name for the administrative domain assigning the identifier. The left-hand side of the at-sign 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.

The identifier may be assigned by the user or by any component of the system along the path. Although Internet mail standards provide for a single identifier, more than one is sometimes assigned.

### [3.4](#) Identity Referencing Convention

In this document, fields references to identities are labeled in a two-part, dotted notation. The first part cites the document defining the identity and the second defines the name of the identity. Hence, <[RFC2822](#).From> is the From field in an email content header, and <[RFC2821](#).MailFrom> is the address in the SMTP "Mail From" command.

## [4.](#) Protocols and Services

NOTE: A discussion about any interesting system architecture is often complicated by confusion between architecture versus implementation. An architecture defines the conceptual functions of a service, divided into discrete conceptual modules. An implementation of that architecture may combine or separate architectural components, as needed for a particular operational environment. It is important not to confuse the engineering

decisions that are made to implement a product, with the architectural abstractions used to define conceptual functions.

Modern Internet email architecture distinguishes four types of functional components, arranged to support a store-and-forward service architecture:

+-----+





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.

#### [4.1](#) Service Components

##### [4.1.1](#) 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.

At the origination side of the service, the oMUA is used to create a message and perform initial "submission" into the transfer infrastructure, via a Mail Submission Agent (MSA). It may also perform any creation- and posting-time archival. An MUA outbox is part of the origination-side MUA.

The recipient-side rMUA works on behalf of the end-user recipient to process received mail. This includes generating user-level return control messages, display and disposition of the received message, and closing or expanding the user communication loop, by initiating replies and forwarding new messages.

An MUA may, itself, have a distributed architecture, such as implementing 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 special class of MUA performs message re-posting, as discussed in the <Mediator> section.

Identity fields set by the MUA include:

[RFC2822](#).From

Actor: Originator

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

[RFC2822](#).Reply-To

Actor: Originator

If a message recipient sends a reply message that would otherwise use [RFC2822](#).From field address(es) 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

Actor: Source

This specifies the address responsible for submission into the transfer service. For efficiency, this field should 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 "notifications" (or "bounces") address, contained in [RFC2821](#).MailFrom -- is made by the [RFC2822](#).Sender. Typically the notifications address is the same as the Sender address. However some usage scenarios require it to be different.

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

Actor: Recipient

These specify MUA recipient addresses. 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

Actor: Recipient

A message might be copied to an addressee whose participation is not to be disclosed to the [RFC2822](#).To or [RFC2822](#).CC 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 single recipient receiving the copy. This usually ensures that even other BCC recipients do not know of each other. 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 or the Originator.

#### [4.1.2](#) Mail Submission Agent (MSA)

A Mail Submission Agent (MSA) accepts the message submission from the oMUA and enforces the policies of the hosting network and the requirements of Internet standards. Enforcement might be passive, involving review and approval or rejection, or it might be active, involving direct modification of the message. An MSA implements a server function to MUAs and a client function to MTAs (or MDAs).

Examples of MSA-styled functions, in the world of paper mail, might range across the very different capabilities of administrative assistants, postal drop boxes, and post office front-counter employees.

The MUA/MSA interface can be implemented within single host and use private conventions for their interactions. Historically, standards-based MUA/MSA interactions have used SMTP [[RFC2821](#)]. However a recent alternative is SUBMISSION [[RFC2476](#)]. Although SUBMISSION derives from SMTP, it operates on a separate TCP port, and will typically impose distinct requirements, such as access authorization.

Identities set by the MSA include:

[RFC2821](#).HELO or [RFC2821](#).EHLO

Actor: Source

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

[RFC2821](#).MailFrom

Actor: 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](#).Rcpt-To

Actor: Recipient

This specifies the MUA inbox 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 address, while the [RFC2821](#).Rcpt-To address specifies a member of that list.

#### [RFC2821](#).Received

Actor: Source

An MSA may record a Received header field, to indicate initial submission trace information, including originating host and MSA host domain names and/or IP Addresses.

### [4.1.3](#) Mail Transfer Agent (MTA)

An <MTA> relays a message to another other MTA or to an <MDA>, in a point-to-point exchange. Relaying is performed by a sequence of MTAs, until the message reaches its destination MDA. Hence an MTA implements both client and server MTA functionality.

The basic functionality of an MTA is similar to that of a packet switch or IP router. That is, it does email store-and-forward email, with a routing decision determining where the next-hop destination shall be. The primary "routing" mechanism for Internet mail is the DNS MX record [[RFC1035](#)]. As with most "link layer" mechanisms Internet mail's SMTP supports a basic level of reliability, by virtue

of providing for retransmission after a transfer failure. However the degree of persistence by an MTA can be highly variable.

However email objects are typically much larger than the payload of a packet or datagram, and the end-to-end latencies are typically much higher. Contrary to typical packet switches (and Instant Messaging services) Internet mail MTAs typically store messages in a manner that allows recovery across services interruptions, such as host system shutdown.

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](#)]

An important characteristic of MTA-MTA communications, over the open Internet, is that they do not require prior arrangement between the

independent administrations operating the different MTAs. Given the importance of spontaneity and serendipity in the world of human communications, this lack of prearrangement, between the participants, is a core benefit of Internet mail and remains a core requirement for it.

Identities set by the MTA include:

[RFC2821](#).HELO

Actor: Relay

The MTA may specify its hosting domain identity for the SMTP HELO or EHLO command operation.

[RFC2821](#).Return-Path

Actor: Source

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

[RFC2822](#).Received

Actor: Relay

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

#### [4.1.4](#) Mail Delivery Agent (MDA)

The <MDA> delivers email to the recipient's inbox.

A Mail Delivery Agent (MDA) 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 Administrative Domain, such as at an organizational border gateway. However it is required for the MDA, if only because the MDA must know where to store the message. This knowledge is used to achieve differential handling of messages.

Using Internet protocols, delivery is effected with POP [[RFC1939](#)] or IMAP [[RFC3501](#)]. When coupled with an internal, local mechanism, SMTP permits "push" delivery to the recipient system, at the initiative of the upstream email service. POP is used for "pull" delivery at the initiative of the recipient system. Notably, SMTP and POP effect a

transfer of message control from the email service to the recipient host. In contrast, IMAP provides on-going, interactive access to a message store, and does not effect a transfer of message control to the end-user host. Instead, control stays with the message store host that is being access by the user.

Identities set by the MDA include:

[RFC2821](#).HELO or [RFC2821](#).EHLO

Actor: Relay

The MDA may specify its hosting domain identity for the SMTP HELO or EHLO command operation.

[RFC2822](#).Received

Actors: Source, Relay, Dest

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

#### [4.1.5](#) Message Store

An MUA's uses a long-term Message Store (MS). A rich set of choices for the use of that store derives from permitting more than one to be associated with a single user, demonstrated as MS-1 and MS-2 in the Figure. MS-1 is shown as being remote from the MUA and MS-2 as being local. Further the relationship between two message store may vary. Between the MDA and the MUA, these choices are supported by a wide variety of protocol options.

The operational relationship among two 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 moved from any remote store, rather than (also) being retained there.

Disconnected: A remote MS and a local MS synchronize all or parts of their contents, while connected. The user may make changes while disconnected, and the two stores are re-synchronized upon reconnection.

## [4.2](#) Operational Configuration

Mail service components can be arranged into numerous organizational structures, each with independent software and administration. One common arrangement is to distinguish:

1. an open, core, global email transfer infrastructure
2. independent transfer services in networks at the edge of the core
3. end-user services

Edge networks may use proprietary email standards. However the distinction between "public" 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.

On the other hand, real-world operations of Internet mail environments do impose boundaries such as access control at organizational firewalls to the Internet. It should be noted that the current Internet Mail architecture offers no special constructs for these configuration choices. The current design of Internet mail is for a seamless, end-to-end store-and-forward sequence. It is possible that the architectural enhancement will not require new protocols, but rather will require clarification of best practises, as exemplified by a recent effort [[ID-spamops](#)]

#### [4.3](#) Layers of Identity References

For a message in transit, the core identity fields combine into:

+-----+-----+-----+			
Layer	Field	Set By	
+-----+-----+-----+			
Message Content	MIME Header	Originator	



Message header fields	From	Originator
	Sender	Source
	Reply-To	Originator
	To, CC, BCC	Originator
	Received	Source, Relay, Dest
	Return-Path	MDA, from MailFrom
SMTP	HELO	Latest Relay Client
	MailFrom	Source
	RCPT-TO	Originator
IP	IP Address	Latest Relay Client

## [5.](#) Message Data

### [5.1](#) Envelope

Information that is directly used or produced by the email transfer service 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](#)].

Direct envelope addressing information, as well as optional transfer directives, are carried in-band by MTAs. All other envelope information, such as trace records, is carried within the content header fields. Upon delivery, SMTP-level envelope information is typically encoded within additional content header fields, such as Return-Path and Received (From and For).

### [5.2](#) Message Header Fields

Header fields are attribute/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. A complete set of registered header fields is being developed through [[ID-hdr-reg](#)].

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

### [5.3](#) Body

The body of a message might simply be lines of ASCII text or it might be structured into a composition of multi-media, body-part attachments, using MIME [[RFC2045](#)], [[RFC2046](#)], [[RFC2047](#)], [[RFC2048](#)], and [[RFC2049](#)]. It should be noted that MIME structures each body-part into a recursive set of MIME header field meta-data and MIME Content sections.

## [6.](#) Two Levels of Store-And-Forward

Basic email transfer is accomplished with an asynchronous store-and-forward communication infrastructure. This means that moving a message from an originator to a recipient involves a sequence of independent transmissions through some number of intermediaries, called MTAs. A very different task is the user-level process of re-posting a message through a new submission process, after final delivery for an earlier transfer sequence. Such MUA-based re-posting shares some functionality with basic MTA relaying, but it enjoys a degree of freedom with both addressing and content that is not available to MTAs.

The primary "routing" mechanism for Internet mail is the DNS MX record [[RFC1035](#)]. It is an advertisement, by a recipient domain, of hosts that are able to relay mail to it, within the portion of the Internet served by this instance of the DNS.

### [6.1](#) MTA Relaying

MTAs relay mail. They are like packet-switches and IP routers. Their job is to make routing assessments and to move the message payload data closer to the recipient. It is not their job to reformulate the payload or to change addresses in the envelope or the content.

### [6.2](#) MUA Forwarding

As discussed in <Mediator> section, forwarding is performed by MUAs that take a received message and submit it back to the transfer service, for delivery to one or more different addresses. A forwarded message may appear identical to a relayed message, such as for Alias forwarders, or it may have minimal similarity, as with a Reply.

#### [6.2.1](#) MUA Basic Forwarding

The simplest type of forwarding involves creating an entirely new

message, with new content, that includes the original message between

Originator-1 and Recipient-1. However this forwarded communication is between Recipient-1 (who could also be called Originator-2) and a new recipient, Recipient-2. The forwarded message is therefore independent of the original message exchange and creates a new message dialogue.

### [6.2.2](#) MUA Re-Sending

A recipient may wish to declare that an alternate addressee should take on responsibility for a message, or otherwise become involved in the original communication. They do this through a user-level forwarding function, called re-sending. The act of re-sending, or re-directing, splices a communication between Originator-1 and Recipient-1, to become a communication between Originator-1 and new Recipient-2. In this case, the content of the new message is the old message, including preservation of the essential aspects of the original message's origination information.

Identities specified in a resent message include

#### [RFC2822](#).From

Actor: 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

Actor: Originator

If this field is present in the original message, it should be retained in the Re-sent message.

#### [RFC2822](#).Sender

Actor: Source

This field is expected to contain the original Sender value.

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

Actor: Recipient

Crocker

Expires July 27, 2005

[Page 24]

---

Internet-Draft

E-Mail Architecture

January 2005

These specify the original message recipients.

[RFC2822](#).Resent-From

Actor: Mediating 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

Actor: Mediating Source

The address of the agent responsible for re-submitting the message. For efficiency, this field should be omitted if it contains the same address as [RFC2822](#).Resent-From. However this does not mean there is no Resend-Sender specified. Rather, it means that that header field is virtual and that the address in the Resent-From field must be used. Specification of the error return addresses (the "bounces" address, contained in [RFC2821](#).MailFrom) is made by the Resent-Sender. Typically the bounce address is the same as the Resent-Sender address. However some usage scenarios require it to be different.

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

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

[RFC2821](#).MailFrom

Actor: Mediating Source

The agent responsible for re-submission ([RFC2822](#).Resent-Sender) is also responsible for specifying the new [RFC2821](#).MailFrom address.

[RFC2821](#).Rcpt-to

Actor: Recipient

This will contain the address of a new recipient

[RFC2822](#).Received

Crocker

Expires July 27, 2005

[Page 25]

---

Internet-Draft

EMail Architecture

January 2005

Actor: Mediating Source

When re-sending a message, the submission agent may record a Received header field, to indicate the transition from original posting to resubmission.

### [6.2.3](#) MUA Reply

When a recipient formulates a response to a message, the new message is not typically viewed as being a "forwarding" of the original.

### [6.2.4](#) MUA Gateways

Gateways perform the basic routing and transfer work of message relaying, but they also make any message or address modifications that are needed to send the message into the next messaging environment. 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 may have significant administrative differences, it is easy to think that a gateway is merely an MTA. The critical distinguish between an MTA and a gateway is that the latter modifies addresses and/or message content.

A gateway may set any identity field available to a regular MUA.

Identities typically set by gateways include:

[RFC2822](#).From

Actor: 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 may translate addresses in whatever way will allow them continue to be useful in the target environment.

[RFC2822](#).Reply-To

Actor: 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

Actor: Source

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

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

Actor: Recipient

These usually retain their original addresses.

[RFC2821](#).MailFrom

Actor: Source

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

## [RFC2822](#).Receive

Actors - Source, Relay, Dest

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

### [6.2.5](#) MUA Alias Handling

A simple re-addressing facility that is available in most MDA implementations is called Aliasing. It is performed just before placing a message into the specified recipient's inbox. Instead, the message is submitted back to the transfer service, for delivery to one or more alternate addresses. Although implemented as part of the message delivery service, this facility is strictly a recipient user function. In effect it resubmits the message to a new address, on behalf of the listed recipient.

What is most distinctive about this forwarding mechanism is how closely it compares to normal MTA store-and-forward. In reality its only interesting difference is that it changes the [RFC2821](#).RCPT-TO value. Notably it does not typically change the [RFC2821](#).Mailfrom

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

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

Actor: Recipient

These retain their original addresses.

[RFC2821](#).Rcpt-To

Actor: Recipient

This field contains an alias address.

## [RFC2821](#).MailFrom

Actor: Mediating Source

The agent responsible for submission to an alias address will usually retain the original address to receive handling notifications. The benefit of retaining the original MailFrom value is to ensure that the origination-side agent knows of 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

Actor: Mediating Recipient

The agent should record Received information, to indicate the delivery to the original address and submission to the alias address. The trace of Received header fields should include everything from original posting through final delivery to the alias.

### [6.2.6](#) MUA Mailing Lists

Mailing lists have explicit email addresses and they forward messages to a list of subscribed members. Mailing list processing is a user-level activity, outside of the core email transfer service. The mailing list address is, therefore, associated with a distinct user-level entity that can perform arbitrary actions upon the original message, before submitting it to the mailing list membership. Hence, mailing lists are similar to gateways.

Identities set by a mailing list processor, when submitting a message, include:

## [RFC2919](#).List-id

Actor: Mediating Originator

This provides a global mailing list naming framework that is



independent of particular hosts. Although [[RFC2919](#)] is a standards-track specification, it has not gained significant adoption.

#### [RFC2369](#).List-\*

Actor: Mediating Recipient

[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.

#### [RFC2822](#).From

Actor: Originator

Names and email addresses for the original author(s) of the message content are specified.

#### [RFC2822](#).Reply-To

Actor: Originator

Mailing lists have introduced an ambiguity for the Reply-To field. Some List operations choose to force all replies to go to all list members. They achieve this by placing the list address into the [RFC2822](#).Reply-To field. Hence, direct, "private" replies only to the original author cannot be achieved by using the MUA's typical "reply to author" function. If the author created a Reply-To field, its information is lost.

#### [RFC2822](#).Sender

Actor: 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.

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

Actor: Mediating Recipient

These will usually contain the original list of recipient addresses.

[RFC2821](#).MailFrom

Actor: Mediating Source

This may contain the original address to be notified of transmission issues, or the mailing list agent may 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 notifications.

[RFC2821](#).Rcpt-To

Actor: Recipient

This contains the address of a mailing list member.

[RFC2821](#).Received

Actor: Mediating Recipient

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

## 7. Security Considerations

This document does not specify any new Internet mail functionality. Consequently it should introduce no new security considerations.

However its discussion of the roles and responsibilities for different mail service modules, and the information they create, highlights the considerable security considerations that must be present when implementing any component of the Internet mail service.

Internet-Draft

EMail Architecture

January 2005

## 8 References

### [ID-hdr-reg]

"Registration of mail and MIME header fields",  
[draft-klyne-hdrreg-mail-04.txt](#) (work in progress), Apr 2004.

### [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.

[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.

[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.

[RFC2048] Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", [BCP 13](#), [RFC 2048](#), November 1996.

[RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail

Crocker

Expires July 27, 2005

[Page 31]

---

Internet-Draft

E-Mail Architecture

January 2005

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.

[RFC2298] Fajman, R., "An Extensible Message Format for Message Disposition Notifications", [RFC 2298](#), March 1998.

[RFC2304] Allocchio, C., "Minimal FAX address format in Internet Mail", [RFC 2304](#), March 1998.

[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.

[RFC2421] Vaudreuil, G. and G. Parsons, "Voice Profile for Internet Mail - version 2", [RFC 2421](#), September 1998.

[RFC2423] Vaudreuil, G. and G. Parsons, "VPIM Voice Message MIME Sub-type Registration", [RFC 2423](#), September 1998.

[RFC2442] "The Batch SMTP Media Type", [RFC 2442](#), November 1998.

[RFC2476] Gellens, R. and J. Klensin, "Message Submission", [RFC 2476](#), December 1998.

[RFC2645] "On-Demand Mail Relay (ODMR) SMTP with Dynamic IP Addresses", [RFC 2465](#), August 1999.

[RFC2782] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), February 2000.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", [RFC 2821](#),

April 2001.

- [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.
- [RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service

Crocker

Expires July 27, 2005

[Page 32]

---

Internet-Draft

E-Mail Architecture

January 2005

Extension for Delivery Status Notifications (DSNs)", [RFC 3461](#), January 2003.

- [RFC3501] Crispin, M., "Internet Message Access Protocol - Version 4rev1", [RFC 3501](#), March 2003.

#### Author's Address

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

Phone: +1.408.246.8253  
E-Mail: [dcrocker@bbiw.net](mailto:dcrocker@bbiw.net)

#### [Appendix A](#). Acknowledgements

The Email Architecture section derives from [draft-hutzler-spamops \[ID-spamops\]](#). The text has been further elaborated.

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 early drafts. Additional

review and suggestions were provided by Nathaniel Borenstein, Ed Bradford, Cyrus Daboo, Tony Finch, Ned Freed, Eric Hall, Bruce Lilly, Eric Hall, Chris Newman, Jochen Topf.

#### Intellectual Property Statement

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

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

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement

this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Disclaimer of Validity

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 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.

#### Copyright Statement

Copyright (C) The Internet Society (2005). 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.

#### Acknowledgment

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