InterPARES 3 Project
International Research on Permanent Authentic Records in Electronic Systems
TEAM Italy

| | |
|---|---|
| **Title:** | **General Study 05 – Keeping and Preserving E-mail** |
| **Status:** | Final (public) |
| **Version:** | 4.1 |
| **Submission Date:** | September 2008 |
| **Last Revised:** | May 2009 |
| **Release Date:** | June 2009 |
| **Author:** | The InterPARES 3 Project, TEAM Italy |
| **Writer(s):** | Gianfranco Pontevolpe<br>Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA)<br>Silvio Salza<br>Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma “La Sapienza” |
| **Project Component:** | Research |
| **URL:** | http://www.interpares.org/ip3/display_file.cfm?doc=ip3_italy_gs05a_final_report.pdf |

# Document Control

| Version history | | | |
|---|---|---|---|
| Version | Date | By | Version notes |
| 1.0 | 2008-09-23 | G. Pontevolpe, S. Salza | Discussion draft prepared following identification of general study at the first InterPARES 3 Project, International Summit, Rome, Italy, 1-3 October 2007. |
| 2.0 | 2008-12-30 | G. Pontevolpe, S. Salza | Draft version prepared following incorporation of draft 1.0 feedback received during the third InterPARES 3 Project, International Summit, Mexico City, Mexico, 1-3 October 2008. |
| 2.1 | 2009-01-03 | R. Preston | Minor content and copy editing to sections 1-4. |
| 3.0 | 2009-02-18 | R. Preston | Major content editing to sections 5-7. |
| 4.0 | 2009-05-29 | G. Pontevolpe, S. Salza | Final public version prepared by the authors, with major editing, and taking into account some of the comments and editing suggestions in v. 2.1 and 3.0. |
| 4.1 | 2009-05-23 | R. Preston | Minor copy editing; addition of Lists of Figures and Tables. |

# Keeping and preserving e-mail

GIANFRANCO PONTEVOLPE [†] AND SILVIO SALZA [†‡]

pontenvolpe@cnipa.it
salza@dis.uniroma1.it

*This report was produced by CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione), the Agency of the Italian Government for ICT infrastructures in the Italian Public Administration, in cooperation with the InterPARES 3 Project, and is part of an activity aimed at proposing guidelines for the management and preservation of e-mail messages as records in the Italian Public Administration.*

† CNIPA Centro Nazionale per l'Informatica nella Pubblica Amministrazione, Rome, Italy, www.cnipa.gov.it

‡ Università degli Studi di Roma "La Sapienza," Dipartimento di Informatica e Sistemistica, Rome, Italy, http://www.dis.uniroma1.it/~salza/

## Table of Contents

## List of Figures

## List of Tables

## Note to the reader

The aim of this study is to investigate the technical aspects relevant to the e-mail creation, capture and maintenance (i.e., records management) and permanent preservation (i.e., archival) processes. This is important since e-mail messages are a very peculiar kind of electronic document, with a rather complex structure, and because of the need to take into account, to some extent, also the peculiar infrastructure through which they are delivered (i.e., the Internet). To achieve this goal we have considered both the functionalities of the commercial products for e-mail management, including the so-called “e-mail archiving” systems, and the requirements expressed in several important reference documents. Devising precise and systematic procedures for e-mail records management and/or permanent preservation is not a goal of this document, and indeed cannot be done in a sufficiently general case, since these procedures may heavily depend on the characteristics of the organization where the process is taking place. For this reason the definition of a more detailed e-mail records management and permanent preservation model should be carried out as a separate task, within the InterPARES 3 Project, and deserves a more thorough discussion, involving records management, archival and IT competences.

> *It soon became obvious that the ARPANET was becoming a human-communication medium with very important advantages over normal U.S. mail and over telephone calls. One of the advantages of the message systems over letter mail was that, in an ARPANET message, one could write tersely and type imperfectly, even to an older person in a superior position and even to a person one did not know very well, and the recipient took no offense. The formality and perfection that most people expect in a typed letter did not become associated with network messages, probably because the network was so much faster, so much more like the telephone*.
>
> J.C.R. Licklider, 1978

## 1 Introduction

The first e-mail was sent in 1971 between two computers that were sitting side-by-side in the same room, but it went through the ARPAnet (the ancestor of the Internet). It was the first time a message was sent across a computer network in a systematic way.

The impressing remark by J.C.R. Licklider, which we have quoted above, came just a few years later, when e-mail was still restricted to a limited milieu in the scientific community, and the widespread use of it was at least a decade ahead. Licklider, an MIT psychologist who formulated the earliest ideas of a global computer network and greatly contributed to the ARPAnet, had indeed a very neat view of what was to come, and a prophetic feeling about the role that the new medium might play in human communication.

Presently, e-mail is by far the most widely used form of written communication; it has been estimated that more than 100 billion e-mails are sent daily, and that the number will reach 300 billion by 2010. Moreover, in the last decade it has become more and more evident that in all business, government and even private activities, a crucial share of the relevant information is exchanged through e-mail messages, and that, in most cases, that information can be found *only* in the e-mail and nowhere else. For instance, it has been estimated that e-mail represents about 75% of corporate intellectual property.[1]

The need for managing and preserving e-mail has therefore become evident: it would not be wise to manage and preserve the other documents and miss the e-mail, where we know that the largest share of information is concentrated. Not surprisingly, therefore, in the last few years, many corporations and government agencies devoted a substantial amount of effort to e-mail management, and this has triggered a market that is expected to reach in 2008 half a billion dollars in software licenses and maintenance services.

[1] Peter Gerr (2004), "IntelliResearch looks to shake up a crowded Enterprise Message Archiving (EMA) market," Enterprise Strategy Group. Available at http://www.enterprisestrategygroup.com/Login.asp?frompage=BuyInsight.asp&ReportID=216.

A more detailed analysis reveals several key motivations driving e-mail archiving activities.

*Storage concerns*

The volume of e-mail messages that corporations and large organizations must handle is very large and growing fast. On the other hand e -mail servers have not been designed to store and manage a large amount of messages and attachments for long periods of time.

As a consequence, most organizations enforce size limits to their employees' mailboxes. This often leads users to routinely backup the messages *they* consider *relevant* on their own PCs, before the messages disappear from the servers. The whole procedure is, of course, informal, uncontrolled and unreliable. Moreover the backed-up messages can only be accessed by the individual users who have stored them (if they are still able to find them).

Up to now, overcoming storage concerns is still the main motivation to e-mail archiving, and hence the strongest market driver.

*Strategic relevance*

E-mail messages have become an increasingly important and strategic resource for the organizations and, hence, should be centrally managed and selected for maintenance and preservation according to precise and well defined criteria. This helps to automate and accelerate business processes, and may produce substantial savings by cutting the time spent in locating and retrieving messages.

Moreover, when an archival solution is deployed, e-mail messages can be integrated with other organization data and analyzed to monitor business processes and to extract knowledge that can help support business strategies.

*Regulatory compliance*

Most companies have been recently fined large amounts of money for failing to maintain corporate e-mail records. In the most evident case, Morgan Stanley was fined in 2005 $1.45 billion, in a ca se dubbed by some as 'legal Chernobyl,' for being unable to produce corporate e-mail records—i.e., for failing to reproduce e-mail requested under investigation (back-up tapes lost or unrecoverable).[2] Smaller amounts of money have been awarded in other cases, but the overall figure has totaled in the last few years several billions of dollars.

In the United States, according to new Federal Rules of Civil Procedure Amendments, the production of electronic information is no longer optional.[3] American companies should therefore be prepared to support electronic discovery and be able to exhibit in a very short time all records requested by a Court—chiefly e-mails (which have played a central role in many recent cases). Although the most evident cases concern private organizations, government agencies have to comply as well.

[2] Landon Thomas, Jr., "Jury Tallies Morgan's Total at $1.45 Billion," *The New York Times*, May 19, 2005. Available at http://www.nytimes.com/2005/05/19/business/19perelman.html?_r=1&th&emc=th.

[3] See http://www.uscourts.gov/rules/civil2007.pdf.

Regulatory compliance has triggered, in the last few years, many organizations to set-up e-mail archiving systems, and it is in the United States a very strong market driver.

*Historical preservation*

Last, but not least, e-mail messages with archival value should be preserved permanently as historical records, in the interest of future generations. This is especially true since, as we have already remarked, e-mail has become the most important form of communication between individuals, replacing paper-based correspondence and, in many cases, substituting for or integrating telephone conversations.

Historians of future generations may have a better chance to investigate the Internet age than the previous part of the twentieth century when all quick communication went through the telephone wires, leaving almost no tangible records to be preserved in the archives. From our side, we should feel the responsibility of preserving such valuable information.

The purpose of this report is to give a concise but complete account of the main problems connected to e-mail records management and permanent preservation, point out the main issues and draw up the basic policies and procedures. This is no trivial task, since e-mail messages are a very peculiar kind of digital document, with a rather complex structure, and because of the need to take into account, to some extent, also the peculiar infrastructure through which they are delivered (i.e., the Internet).

Therefore, we include in the report an overview about the e-mail infrastructure and the message format, issues that some users may consider unpleasant technicalities, but that we believe are essential to correctly understand some of the problems connected to the management and preservation of e-mail messages.

The report is organized as follows:

Section 2 deals with the Internet e-mail infrastructure. We briefly show how e-mail works and how end users have access to it, and discuss the main Internet standards related to e-mail that have been designed to guarantee the interoperability of heterogeneous systems over the network.

Section 3 discusses the format and the structure of e-mail messages, a very important aspect, since the format of a digital document is a central issue in the maintenance and permanent preservation processes. Moreover we show how important information can be found in the message to be extracted as metadata.

Section 4 is devoted to security issues—i.e., the vulnerabilities that arise in relation to the Internet, through which messages are delivered in an uncontrolled environment—the consequent problems in assessing message authenticity, and the privacy and confidentiality issues.

Section 5 deals with the heart of the problem, i.e. managing and preserving e-mail, a complex process which may go from just maintaining e-mails as transitory records within a transient storage system, to classifying, registering and maintaining corporate e-mail records and their metadata in a dedicated corporate recordkeeping system (e.g., electronic records management system, or ERMS). Section 5.1 discusses the different strategies that can be used to capture e-mail messages, which can be either based on automated procedures, or may involve the cooperation of the user (i.e., the sender or the recipient of the message). Section 5.2 addresses the crucial problem of preservation formats, which must ensure that the content, the structure and the appearance of the message are preserved, allowing future users to access the information in the message in its original documentary form. Next, problems related to message metadata extraction and to checking and maintaining message authenticity are discussed in sections 5.3 and 5. 4. Finally, issues connected to long-term maintenance and permanent preservation are discussed in section 5.5.

Section 6 discusses the problem of granting users with efficient access to e-mail records, including adequate search and discovery capabilities, while still protecting the records from unauthorized access and accidental or fraudulent manipulation or destruction.

Section 7 analyzes commercial products for e-mail management. We consider e-mail servers, integrated systems and e-mail “archiving” systems, and we analyze the basic and advanced functionalities of the products (both proprietary and open source) with the largest diffusion on the market at present.

Complementary material is presented in Appendix A, in which we briefly discuss the main standards and reference documents containing requirements for the management of digital records that we have taken into account in writing this report. The purpose of the appendix is to compare the approaches in these documents, and to give the reader an access guide to the requirements that specifically concern e-mail management systems.

# 2 The Internet e-mail infrastructure

## 2.1 How does e-mail work?

E-mail is a store-and-forward method of exchanging messages on the Internet. This means that a message sent by a user goes through an asynchronous process of delivery, typically involving a series of steps. In each step the message is stored by an intermediate server on the network, to be forwarded at a later time, until it finally reaches its destination. Timing depends on the availability of connections on the network.

A schema of the delivery process is shown in Figure 1. The process involves a *sender*, say Alice, and a *destination*, say Bob. Both Alice and Bob use specific applications, called *e-mail clients*, running on their PC to send and receive e-mail. Clients do not communicate directly, but have to connect to *e-mail servers* (i.e., special applications run by Alice’s and Bob’s organizations or ISPs) that actually take care of carrying on the message delivery.

The process goes through the following steps:

- Alice composes the message using her *e-mail client*;
- the message is formatted by Alice’s *e-mail client* in a specific *internet e-mail format*, and then is sent to her local *e-mail server*;

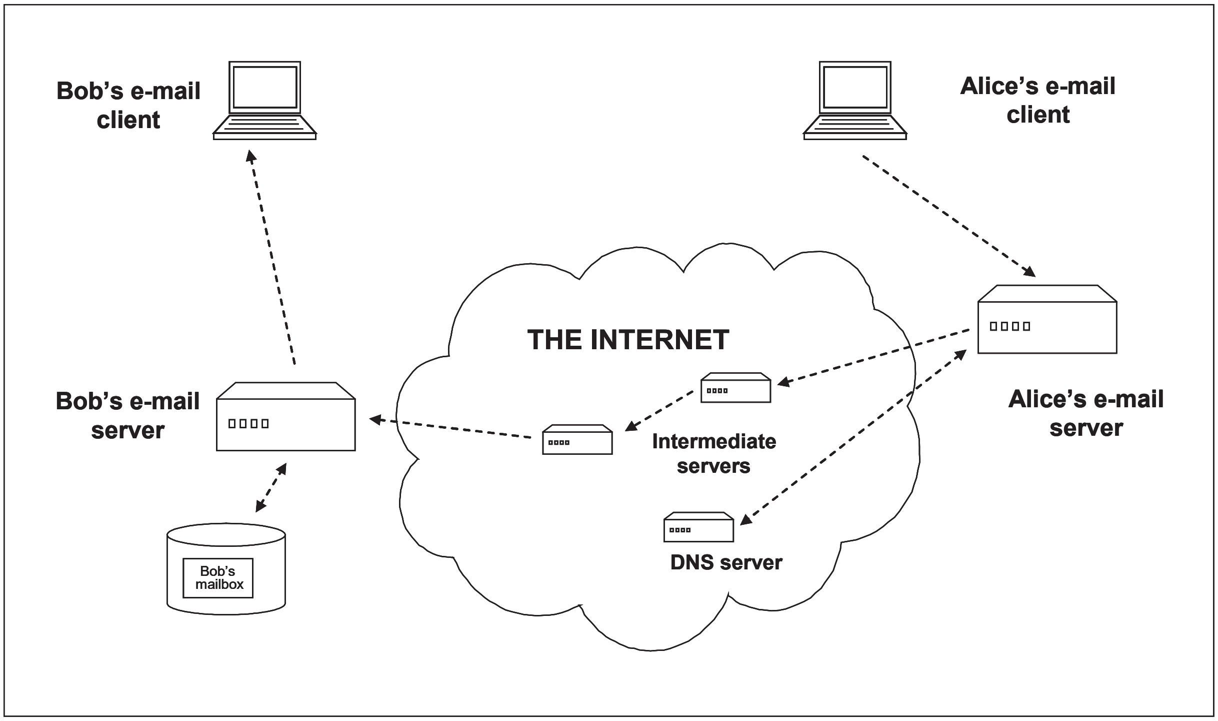

**Figure 1.** Basic e-mail infrastructure

- Alice's *e-mail serve*r locates the address of Bob's *e-mail server*, exploiting the *Domain Name System (DNS)* (i.e., the distributed directory of the Internet);
- the two *e-mail servers* exchange the message, which may go through a series of intermediate servers on the network, and is finally stored by Bob's *e-mail server* in Bob's personal *mailbox*; and
- the message is kept in Bob's mailbox until he reads it and/or downloads it using his *e-mail client*.

This procedure is pretty much the same as the one that Alice and Bob follow when they exchange letters. Their local post offices play the same role as local e-mail servers, and the letter delivery may go through additional post offices (intermediate servers). In both cases, the delivery time and delivery itself are not guaranteed.

The Internet is a *best-effort network*, and the message, like any other information crossing the network, to reach its destination has to go through several servers run by independent organizations that take no commitment on the availability and the quality of their service. Hence, the delivery time cannot be predicted, and the message may even get lost on the way.

Regardless, as we shall discuss later in more detail, all clients and servers involved in the delivery process follow a set of strict rules (protocols). This allows the system to trace all relevant events, and to record all this information in a rather detailed report that is appended to the message. Moreover, in case of failure, the server may automatically reattempt the delivery. Likewise, the sender may ask for delivery reports and receipts to gain evidence that the message has been delivered and/or that the recipient has actually read it.

## 2.2 End-user access to e-mail

End users may access the e-mail system in several different ways.

*E-mail client*

This case corresponds exactly to the basic schema we have discussed in the previous section, where the user runs on his PC a special application specifically designed to interact with the e-mail server. E-mail clients are proprietary or open source software, and there is a large variety of them on the market. Besides the basic functions of sending messages and retrieving them form the e-mail server, which are performed according to standard interaction protocols that ensure interoperability, they usually offer user-friendly interfaces and plenty of additional functions to classify and store messages, manage directories and so on. In this schema, messages are usually downloaded and stored on the user's PC, which may not be convenient for nomadic users who need to access their e-mail from several different PCs.

*Webmail*

This is the way most users access e-mail from their home PC, through a service offered by their Internet Service Providers (ISPs), or by third party organizations, like Hotmail or G-mail. In this schema (see Figure 2), the client application, running on the end user PC, is an Internet browser (Explorer, Mozilla or other) that connects to the web server, where a special application (webmail) is running. The web server acts as an intermediate party, and manages the connection with the e-mail server. Moreover, messages are not downloaded to the user PC, but are directly managed and stored on the web server. This gives a significant advantage to nomadic users, since they may access their mail from many different PCs.

*Integrated systems*

This is the typical solution used by most corporations and large organizations, and is based on the idea of integrating e-mail access in a broader 'collaborative' environment that includes additional functions, such as direct messaging, calendaring, contacts and tasks, and support for mobile and web-based access to information, as well as managing message storage on a central server. The most popular products of this kind are Microsoft Exchange and IBM Lotus Domino. Users run on their PCs proprietary client applications (e.g., Microsoft Exchange or Lotus Notes) that connect to the corporate server, which in turn connects to the e-mail server (see Figure 3). To assist nomadic users these systems also have an optional web interface functionally equivalent to webmail, which allows access from a web browser through the Internet, but the primary interface is still the proprietary one that is typically used on the organization's intranet. Although not very general and affected by proprietary elements, this schema needs to be carefully considered, since it accounts for a large share of the market, and since corporations and large organizations are a very significant case for e-mail archiving.

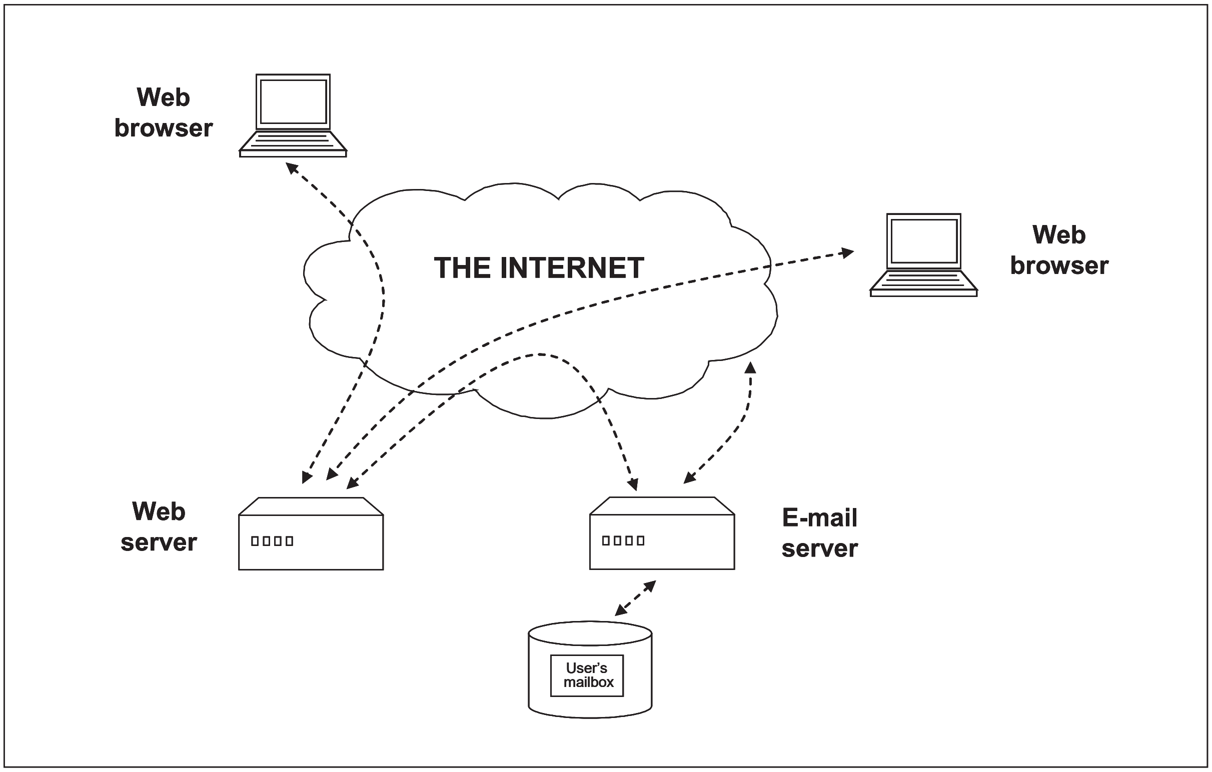

**Figure 2.** Webmail

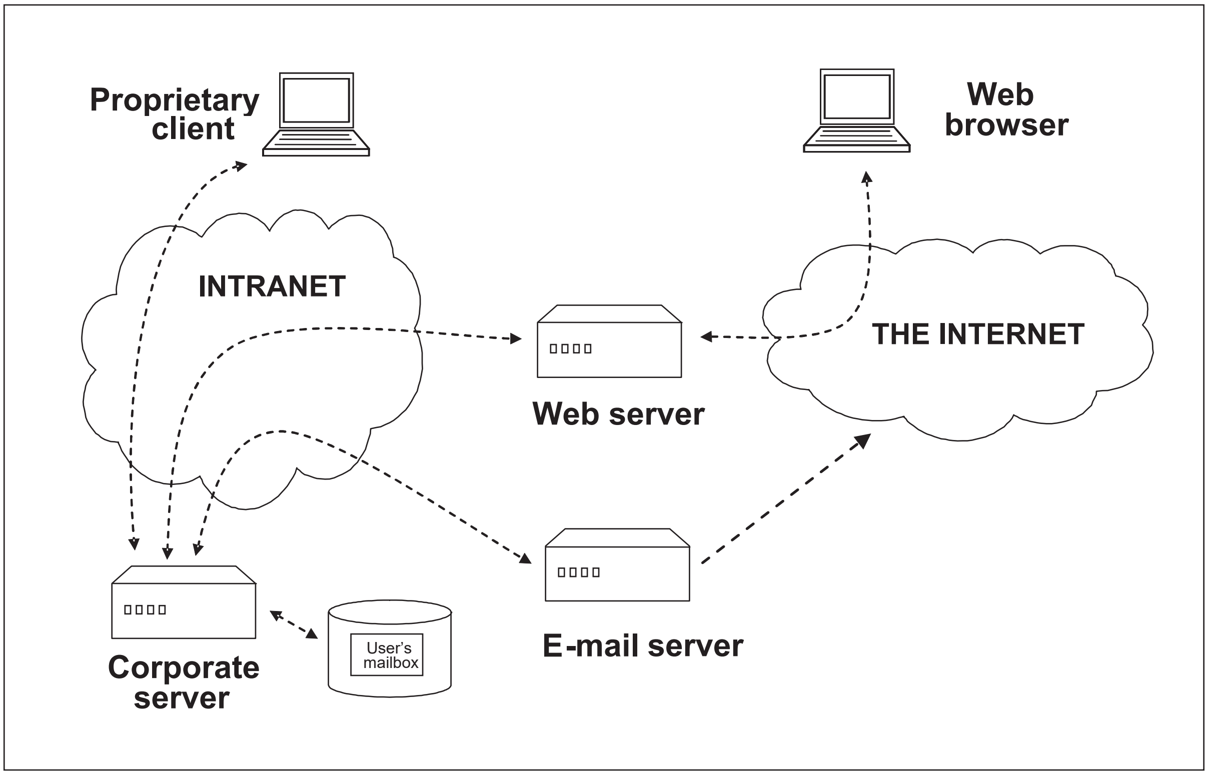

**Figure 3.** Corporate e-mail with integrated system

## 2.3 Interoperability of e-mail systems

As we have seen in the previous sections, exchanging a message involves an interaction among several *agents* (e-mail clients and servers), which are, in general, *heterogeneous* systems—i.e., based on different hardware and software platforms. Moreover, these systems are independently designed and implemented by different parties, potentially without any form of direct coordination.

A main challenge for the Internet e-mail system is, therefore, to ensure *interoperability*—i.e., correct and reliable communication among these heterogeneous systems. Interoperability is based on two main elements:

- *communication protocols*—i.e., sets of rules governing the communication between agents, which ensure that agents may reliably and correctly interact by means of a common language and of standard procedures; and
- *message format*—i.e., a set of formal definitions that specify the structure of the message and how the message and its attachments are encoded; thus, providing for correct interpretation by different e-mail clients, and guaranteeing that the content of the message is correctly rendered to its recipient.

A further requirement is that interoperability must also be guaranteed across time. That means that when the definition of protocols and message format evolve, they should still guarantee backward compatibility (i.e., new rules should still be compatible with old rules). For example, a message formatted according to an old version of the message format standard should be presented correctly by an e-mail client compliant with the new version of the standard.

Unfortunately, this is not always the case and is a major problem to be addressed in e-mail archiving, since we must ensure that the messages that we archive remain readable across time, even if standards evolve.

## 2.4 Internet standards

The standardization process of the Internet is somewhat different from the usual ISO/IEC track; therefore, it is worth introducing how these standards are produced and allowed to evolve.

Internet standards are developed and promoted by the *Internet Engineering Task Force* (*IETF)*,[4] which cooperates closely with the major international standards bodies, including the *International Organization for Standardization (ISO)*,[5] the *International Electrotechnical Commission (IEC)*[6] and the *World Wide Web Consortium (W3C)*,[7] the main international standards organization for the World Wide Web.

The standardization process, which dates back to the early days of the ARPAnet project, is highly cooperative and is based on special documents called *Request for Comments (RFC)*. RFCs are draft documents, mostly proposals of standards, published by IETF and posted on the network as a ‘request for comments.’ Each RFC is assigned

[4] See http://www.ietf.org/.
[5] See http://www.iso.org/iso/home.htm.
[6] See http://www.iec.ch/.
[7] See http://www.w3.org/.

a unique number and is never rescinded or modified. If amendments are needed, a new RFC is issued, with a different number, which supersedes the old one.

As established by RFC 1796, which discusses the standardization process, “not all RFCs are standards.”[8] Some are just memoranda, remarks that people like to share, research papers or preliminary proposals on any matter concerning the Internet and Internet-based systems. Therefore, the IETF assigns to each RFC a *rating*, called *status*.

‘Mature’ RFCs are rated *Standard Track*, and are further divided into *Proposed Standard*, *Draft Standard* and *Internet Standard*. Internet Standards (STD) refer each to an RFC (or a set of RFCs), and are given a unique number, as for the RFCs; however, unlike the RFC number, when the standard evolves, the STD number does not change, but simply refers to a new RFC that supersedes the original one.

## 2.5 Standardization of e-mail transmission

Server-to-server and client-to-server interoperability are ensured by *SMTP, Simple Mail Transfer Protocol*, which is Internet Standard STD 10. SMTP dates back to August 1982 and it is based on RFC 821,[9] and most recently updated by RFC 5321.[10] However, the protocol currently used by the majority of e-mail applications is the one known as *ESMTP* (for *Extended SMTP*) and defined in RFC 2821, published on April 2001.[11]

However, formally, the status of RFC 2821 is still a *Proposed Standard*, and the official standard is still the one defined by RFC 821. This situation of ‘going ahead the official standard,’ is typical of the Internet world, and it is of no use to argue if this is right or wrong: we must just cope with it.

SMTP specifies the way the e-mail client interacts with e-mail servers and delivers the messages to them, and how the e-mail servers (that hence are often called *SMTP-servers*) interact among themselves in such a way that the message goes through several agents and finally reaches its destination. Use of SMTP protocol in the message delivery process is clearly shown in figures 1 and 2.

As far as the problem of e-mail archiving is concerned, this standard is important since it defines the basic format of messages that can be handled by SMTP-servers and that go through the delivery process. This is indeed a very basic format, since only simple text messages in *plain ASCII* (also called 7-bit ASCII or US-ASCII) characters are supported, which is adequate only for English and a few other languages. This limitation is overcome by defining a special way to encode any richer content in plain ASCII characters, to allow the use of a more general set of characters in the message text, and to include in e-mail messages formatted text and multimedia contents, as we shall discuss in section 2.7.

[8] See http://www.apps.ietf.org/rfc/rfc1796.html.
[9] See http://tools.ietf.org/html/rfc821.
[10] See http://tools.ietf.org/html/rfc5321.
[11] See http://www.apps.ietf.org/rfc/rfc2821.html.

## 2.6 Standardization of client-server communication

E-mail clients may retrieve e-mail from servers in several different ways, supported both by standard and proprietary protocols. This is relevant for e-mail archiving, since, according to the different options, messages may either be downloaded from the server and stored on the end-user's PC, or may be kept on the server, which takes them in charge and allows the user to access them at any time. This has a substantial impact, as we shall see later (see section 5.2), on the organization of the message capture process.

Two main protocols are used to read and retrieve e-mail.

- *Post Office Protocol version 3 (POP3).* This protocol (RFC 1939, STD 53)[12] was originally designed to support users with sporadic network connection (i.e., dial-up connection). When the end-user connects he downloads on his PC all the new messages, to conveniently read them offline. Messages are then (usually) deleted by the server. An option exists to leave a copy of the messages on the server, but it is seldom used, since many implementations cannot tell properly between the new messages and the ones that have been already downloaded.
- *Internet Message Access Protocol (IMAP)*. IMAP (RFC 3501, *Proposed Standard*)[13] was specifically designed to meet the needs of nomadic users—i.e., being able to access their e-mail from several different computers. It allows local clients to access mail on a remote server. All messages are stored on the server, where they are kept to be accessed at any time until the user explicitly decides to delete them. Moreover the user may create folders inside his mailbox on the server to organize and archive his messages. Having all the messages on the server is definitely a positive feature for e-mail archiving.

Besides direct client-server connection, POP3 and IMAP are also used as part of the other e-mail access schemes that we have discussed in section 2.2. In webmail the web server uses POP3 or IMAP to retrieve messages from the e-mail server, and SMTP to send messages. Similarly, in integrated systems, the corporate server uses standard protocols to connect to the e-mail server (SMTP server), but proprietary protocols are used in the communication with the end-user client. In both schemes, messages are kept on the server; again, a positive feature for e-mail archiving.

## 2.7 Standardization of message format

The basic format of e-mail messages is defined by RFC 822 (*Format of the Internet Arpa messages*), which is Internet Standard STD 11.[14] RFC 822 dates back to 1982, but most applications can now handle the updated version of message format defined in RFC 2822, which is still formally a *Draft Standard*.[15]

Both RFC 822 and RFC 2822 specify that e-mail messages should contain only *plain ASCII text* (also called 7-bit ASCII or US-ASCII) characters—i.e., characters from the original 128 character ASCII standard code, which dates back to 1963 and was devised

[12] Available at http://tools.ietf.org/html/rfc1939.
[13] Available at http://tools.ietf.org/html/rfc3501.
[14] Available at http://tools.ietf.org/html/rfc822.
[15] Available at http://tools.ietf.org/html/rfc2822.

for plain text English. In fact, as we have seen, SMTP-servers can only handle this type of message. This restriction, in principle, rules out using in e-mail messages other character codes, such as ISO 8859 and Unicode. Hence, for instance, messages should not contain characters with the diacritic marks (accents) used in Latin and north-European languages.

To overcome this limitation, the message format has subsequently been extended by the *Multipurpose Internet Mail Extension (MIME)* standard[16] to support:

- text and headers in character sets other than plain ASCII text;
- messages structured in multiple parts; and
- non-text attachments, including a large variety of multimedia files.

Although universally used and acknowledged by everyone, MIME is not (yet) formally an Internet standard. It is defined by a series of linked documents (RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289), each of whose status is still *Draft Standard* (the last step below *Internet Standard*).

MIME is based on the simple and straightforward idea of encoding non-ASCII characters, and potentially any kind of information attached to the message, with plain ASCII characters. Information on the encoding scheme is added to the message, to allow the decoding of the message when it is retrieved.

All MIME encoding and decoding is performed by e-mail clients when sending and retrieving messages. The message, when transmitted, is made up only of plain ASCII characters; thus, no extension is needed to SMTP and SMTP-servers to handle MIME messages.

MIME is by its own nature extensible, and its definition includes a mechanism to *register* new data types, called *Internet media types* or MIME types, when the need arises. Registration of new data types is managed by the independent *Internet Assigned Numbers Authority (IANA)*, an entity that oversees, among other things, IP address allocation and DNS root management.[17]

A very large number of Internet media types have been registered to date, and this virtually allows attaching to an e-mail message any kind of computer file, notably formatted text, multimedia content, and more. Binary data (pictures, formatted text documents, etc.) are encoded, using a well known schema called BASE64, in plain ASCII characters. Therefore, for instance, a picture will be included in a message as a long sequence of plain ASCII characters.

A further recent extension to MIME is *Secure/Multipurpose Internet Mail Extension (S/MIME)*,[18] which defines a standard for public key encryption and signing of e-mail encapsulated in MIME (see section 4.4).

[16] Available at http://www.mhonarc.org/~ehood/MIME/.
[17] See http://www.iana.org/.
[18] Available at http://www.ietf.org/rfc/rfc3851.txt.

# 3 Format and structure of Internet e-mail messages

Message format and enco ding is of crucial importance in e-mail management and permanent preservation, for several reasons.

First, to archive a message, we first need to determine the message structure and to identify all the elements that compose it:

- *message data*: the sender, the recipients, etc.;
- *delivery information*: e-mail servers that handled the message, date sent, date retrieved, etc.;
- *message text*; and
- *attachments*.

Next, all these elements should be extracted from the message, to help decide, through a delicate and complex process, if the message is going to be archived and how it should be classified (see section 5.3).

Finally, we must decide in which format the message and/or its components should be maintained (see section 5.2).

## 3.1 Message structure

An Internet e-mail message consists of two major sections:

- *header*: a sequence of lines, at the beginning of the message, generated by the sender e-mail client and by the e-mail servers involved in the delivery process; and
- *body*: the rest of the message, which contains the message text in plain ASCII characters and/or a text containing non-ASCII characters, and binary data in plain ASCII encoding.

In the simplest case—i.e., the original message format defined in RFC 822—the message body contains only plain ASCII characters. Such messages are straightforward to handle and can just be archived in their native format, and then read again with no need for any form of decoding.

Unfortunately, most messages use extended ASCII or Unicode characters and have attachments and/or are in html format. In all these cases the message must be in MIME format. Accordingly, we shall concentrate in the following sections on the structure of MIME messages.

## 3.2 Message header

The message header is a sequence of lines, called header lines or simply headers, which are produced by the sender e-mail client and by the e-mail servers on the delivery path. The header is terminated by a blank line. All that follows is the message body.

Only a minor part of the information in the message header is displayed by most e-mail clients. This is rather reasonable, since there is a very large variety of headers, many of them optional, and most users would just be confused by too much detail. However, e-mail clients generally allow users to inspect the complete header, if they like to investigate the message origin and the delivery process.

The most common headers are shown in Table 1. We may divide them in four main categories, based on the e-mail management processes to which data refer, as follows:

- *Identity.* These headers specify the sender and the recipients of the message, and add to the basic elements some more details. For instance, the message is (almost) always given by the sender e-mail server a `Message-ID`, which is an identifier that should be unique (at least for that server), and that can therefore be used to reference the message (e.g., in other messages). Moreover, a `Return-Path` can be specified, if different from the sender address, as an address to which all *bounce messages* (i.e., notifications and answers generated by the message), should be sent. Finally, `Sender` specifies the human or automated agent that is actually sending the message on behalf of the *official* sender (i.e., the one mentioned in the `From` header).
- *Delivery.* These headers collect the details about the delivery process. A `Received` record is added to the message each time the message is handled by a server on the delivery path; the first one being the sender's e-mail server, and the last one the recipient's. A timestamp is associated with each step, specifying the local date/time the message arrived to the receiving server, expressed in the standard format in which the GMT and the time shift are given. Additional headers specify if the sender requested a receipt, and to which address it should be sent. However, different e-mail clients may behave in different ways in handling the receipt information. Therefore, care should be taken in considering the lack of a return receipt as evidence that the message has not been delivered or read.
- *Thread.* These headers are used in messages that are sent in reply to other messages, a very common case, and in messages that are used to forward other messages. These groups of messages form therefore a *thread*, and some of the header information of the message initiating the thread is included in the new message, notably the *message identifier*. Headers referring to threads are of special interest in e-mail archiving, since they allow for the extraction of metadata that connect a message to other messages.
- *MIME.* These headers specify the structure of the message body and the MIME version, which, despite the evolution of the standard, is still 1.0. The `Content-Type` header specifies if the message contains one or several parts; in the latter case, a `boundary` is also specified (i.e., a string that separates the multiple parts of the message in the message body). If instead the message contains a single part, the `Content-Type` and the `Content-Transfer-Encoding` are directly specified in the header.
- *Miscellaneous*. Additional headers may be added that refer to security applications, spam filtering and other e-mail management processes.

**Table 1.** Most common header lines
(A means always present, F frequent, O optional)

### Identity

| HEADER | DESCRIPTION | ORIGIN | PRESENT |
|---|---|---|---|
| `Date:` | *Date/time sent* | Sender client | A |
| `From:` | *Address of sender* | Sender client | A |
| `Sender:` | *Address of sender's assistant* | Sender client | O |
| `Organization:` | *Organization of author* | Sender client | O |
| `To:` | *Address of recipients (may be a list)* | Sender client | O |
| `Cc:` | *Address of recipients in carbon copy* | Sender client | F |
| `Bcc:` | *Address of recipients in blind carbon copy* | Sender client | F |
| `Subject:` | *Message summary* | Sender client | A |
| `Message-ID:` | *Unique identifier assigned by the sender* | Sender server | F |
| `Return-Path:` | *Address for 'bounce messages'* | Sender client | O |

### Delivery

| HEADER | DESCRIPTION | ORIGIN | PRESENT |
|---|---|---|---|
| `User-Agent:` | *Sender e-mail client software* | Sender client | A |
| `Delivered-To` | *Recipient mailbox (may be a list)* | Recipient server | A |
| `Received:` | *One for each step in the delivery path* | Server | A |
| `from` | *Server which sent the message* | Server | A |
| `by` | *Server which received the message* | Server | A |
| `with` | *Server ESMTP identifier* | Server | A |
| `date` | *Date/time received* | Server | A |
| `Return-Receipt-To:` | *Address to send a read receipt* | Sender client | O |
| `Disposition-Notification-To:` | *Address to send a read receipt* | Sender client | O |

### Thread

| HEADER | DESCRIPTION | ORIGIN | PRESENT |
|---|---|---|---|
| `In-Reply-To:` | *Message ID to which the message replies* | Sender client | O |
| `References:` | *Message ID to which the message refers* | Sender client | O |
| `Resent-From:` | *Address of sender forwarding the message* | Sender client | O |
| `Resent-To:` | *Address of the recipient forwarded message* | Sender client | O |
| `Resent-Subject` | *Subject of the forwarding message* | Sender client | O |

### MIME

| HEADER | DESCRIPTION | ORIGIN | PRESENT |
|---|---|---|---|
| `MIME-Version:` | *Always 1.0* | Sender client | A |
| `Content-Type:` | *Specifies content and structure of the body* | Sender client | O |
| `boundary` | *Separator in a multipart messages* | Sender client | O |
| `Content-Transfer-Encoding` | *Encoding scheme* | | |

Altogether, the message header contains crucial information for e-mail archiving. As we shall discuss in section 5.3, most of the message metadata can be extracted from the header. However, this is not a straightforward task, since, as we already mentioned, only a few headers are mandatory, some are used interchangeably, and their order and syntax are quite flexible.

Moreover, the reliability of this information depends on the correctness of the implementation of e-mail clients and servers involved in the delivery process, and these applications are potentially implemented and marketed without any control and certification. However, this leads to a more general problem concerning the Internet e-mail system as a whole, which makes, as we shall discuss in section 4, the forging of an e-mail a rather easy task, and avoiding it a quite complex one.

### 3.3 Message body

A message in MIME format may contain one or several parts.

A single part message is a plain text message with no attachments. The corresponding `Content-Type` in the header is `text/plain`, and it also specifies character encoding. For messages containing only plain ASCII characters, the `Content-Transfer-Encoding` is `7-bit`. Otherwise, if the character set is other than plain ASCII, a different encoding is used, very frequently `quoted-printable`, an encoding scheme that represents directly plain ASCII characters, and encodes ISO 8859 (extended ASCII) or Unicode characters with three plain ASCII characters each. Although this and other encodings are very common, misinterpretation of characters with diacritic marks when reading a message is a rather common e-mail client failure.

A similar encoding scheme, called *Encoded-Word*, is used for textual header information in character sets other than plain ASCII.

The structure of a single part message is represented in Figure 4. This message uses the ISO 8859-1 (Western Europe) encoding, and contains accented characters both in the `Subject` header and in the text.

### 3.4 Multipart messages

A multipart MIME message is composed of several parts separated by a *boundary*—i.e., by the string defined in the top-level `Content-Type` header (see section 3.2)—that is placed between any two parts. The structure can be nested—i.e., any of the part may have a multipart structure itself.

Multipart messages can be of several types, which are specified as *subtypes* in the `Content-Type` header. For our purposes the relevant subtypes are:

- `Multipart/mixed`

  This subtype is intended for packing in a single message several files with different data types, which are specified by the `Content-Type` headers. The default content type is `text/plain`. According to user option settings, e-mail clients may display some of these files online (e.g., pictures) and/or as attachments. This subtype is generally used to send messages with attachments.

```
Message-ID: <006401c91467$186fb1d0$6602a8c0>
From: "Silvio Salza" <salza@dis.uniroma1.it>
To: "Silvio Salza" <salza@dis.uniroma1.it>
Subject: Sample single part message
Date: Fri, 12 Sep 2008 01:35:37 +0200
Organization: =?iso-8859-1?Q?Universit=E0_di_Roma?=
MIME-Version: 1.0
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Message from the University of Rome
Messaggio dall'Universit=E0 di Roma
```

**Figure 4.** Structure of a single part message

The order of parts is meaningful, and should be used by the e-mail clients in displaying them.

- `Multipart/mixed`

  This subtype is intended for packing in a single message several files with different data types, which are specified by the `Content-Type` headers. The default content type is `text/plain`. According to user option settings, e-mail clients may display some of these files online (e.g., pictures) and/or as attachments. This subtype is generally used to send messages with attachments. The order of parts is meaningful, and should be used by the e-mail clients in displaying them.

- `Multipart/alternative`

  This subtype is used to send several "alternative" versions of the same content—i.e., of the same message—the format of each version being specified by its own `Content-Type` header. The alternative parts appear in an order of increasing ‘faithfulness’ to the original content, with the best choice being the last. E-mail clients should recognize that the content of the various parts are interchangeable, and display the ‘best’ type based on their capability and/or on user option settings.

  A very common instance of `Multipart/alternative` are messages that are sent both in plain text (`Content-Type: text/plain`) and in HTML format (`Content-Type: text/html`). The plain text part provides backwards compatibility, while the html part allows use of formatting and hyperlinks.

Therefore, the two parts do not contain exactly the same information, the html part being somewhat richer. A sample message of this kind is shown in Figure 5.

- `Multipart/digest`

  This subtype is syntactically identical to `multipart/mixed`, but the semantics are different. More specifically, in a digest the default `Content-Type` value for a body part is changed from `text/plain` to `message/rfc822`. This media type indicates that the body contains an encapsulated message with the syntax of an RFC 822 message.

  The `Multipart/digest` is used to send collections of messages in a single message and, very often, for e-mail forwarding.

- `Multipart/related`

  This subtype provides a mechanism for representing compound objects consisting of several interrelated parts. Each part of the object is sent as a part of the multipart message. A common instance of this subtype is represented by messages sending a web page, complete with images. The root part contains the HTML document, and uses image tags to reference images stored in the latter parts.

```
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary=‟---separator---"

This is a multi-part message in MIME format.

---separator---
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Message from the University of Rome

---separator---
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

 < message text in html >

---separator---
```

**Figure 5.** Multipart message in text and html

- `Multipart/report`

  This subtype is meant for e-mail reports of any kind. It is generally used for message delivery reports. It has two required parts, plus an optional one. The first part contains a human-readable message with a description of the condition that caused the report to be generated. The second part is machine-parsable and contains an account of the reported message handling event. The optional third part contains the message to which the report relates or part of it, with the purpose of helping in diagnosing problems.

- `Multipart/signed`

  This type is used to send messages with digital signatures. It has two parts: a body part (the message) and a signature part. The digital signature is used to authenticate the whole content of the first part. Many signature types are possible, and there is still some lack of standardization. Moreover, signed messages may also be sent using the mixed multipart schema.

- `Multipart/encrypted`

  This type is used to send encrypted messages. It has two parts. The first part contains the information needed to decrypt the second part. Similar to signed messages, there are different implementations, which are specified in the `Content-Type` of the first part, and there is still a lack of standardization.

### 3.5 MIME media types

A MIME media type is an identifier used in a content type header to specify the nature of the data in the body of a MIME entity—i.e., in the body of a single part message or in a part of a multipart message. MIME media types are often referred to as *Internet media types*, since they are used also in other Internet protocols, mainly in HTTP. Their purpose is to allow the correct interpretation of the message content by specifying the file format of its body and attachments.

The MIME media type mechanism is defined in RFC 2046, and has been carefully designed to be extensible, as it is expected that the set of media types will grow significantly over time. To ensure that the set of Internet media types are developed in an orderly, well-specified and public manner, a registration process has been devised that refers to the Internet Assigned Numbers Authority (IANA) we have mentioned above.

Media types are two-level identifiers that specify a *top-level type* and a *subtype*, with optional additional parameters. RFC 2046 defines seven top-level media types. Five of them are *discreet data types* (i.e., specify the format of a single file), and the remaining two are *composite data types* (i.e., specify the structure of a MIME body composed of multiple parts).

The five top-level discreet media types are:

- `text`

  Textual information. The subtype `text/plain` indicates plain text with no formatting and is intended to be displayed directly, without the intermediation of any special software, aside from supporting the character set, which is specified by a `charset` parameter. For instance:

  ```
  Content-type: text/plain; charset=iso-8859-1
  ```

  indicates a text encoded in the extended ISO/IEC-8859-1 character set commonly referred as *Latin 1, Western European*. Other subtypes are used for enriched text, as `text/html` for HTML files, `text/xml` for XML files and `text/css` for CSS (Cascading Style Sheet) files.

- `image`

  Image data—i.e., any information that requires a graphical display device to be rendered. Registered subtypes include all widely used image types, such as `gif`, `tiff`, `jpeg`, `png`.

- `audio`

  Audio data—i.e., any information that requires audio device information, such as a speaker, to be rendered. The more general subtype is `audio/mpeg,` which refers to MP3 or, in general, to MPEG audio. Other audio data subtypes refer to proprietary formats, such as `audio/x-ms-wma` for Windows Media Audio or `audio/x-wav` for WAW (Waveform audio format).

- `video`

  Indicates a time-varying picture image, possibly with colour and coordinated sound. Standard (IANA registered) subtypes are `video/mpeg` for MPEG-1 video with multiplexed audio, `video/mp4` for MP4 video and `video/quicktime` for Quicktime video. Other subtypes refer to proprietary formats, such as `video/x-ms-wmv` for Windows Media Video.

- `application`

  The application type is used for data that do not fit in any of the other media types. These data need to be processed by some application program to be rendered. There is a very large variety of application subtypes: up to now, IANA has registered about seven hundred subtypes, most of which are vendor-specific, and their identifiers begin with '`vnd.`'. For instance the `application/vnd.ms-excel` subtype is used for Microsoft Excel files.

  Due to the enormous variety, it is impossible to enumerate even a small set of relevant application subtypes.

### 3.6 Media types and dynamic contents

Actually, the situation with media types is more complex, since, besides the IANA-registered media types, there are many subtypes that are widely used and handled by most e-mail clients, but not (yet) registered with IANA.

For instance,

```
Content-Type: application/msword; name="sample.doc"
Content-Description: sample.doc
Content-Disposition: attachment; filename="sample.doc";
   size=99328; creation-date="Tue, 05 Aug 2008 10:08:40 GMT";
   modification-date="Tue, 05 Aug 2008 10:08:40 GMT"
Content-Transfer-Encoding: base64
```

indicates a Microsoft Word attachment, a very frequent case. Moreover, the `Content-Type` definition is completed by several parameters, which specify some object metadata and the encoding, and it is not always evident where to find the related documentation.

Dealing with media types poses several problems when managing and preserving e-mail, as we shall discuss in more detail in section 5.3. Indeed, the media type paradigm has been conceived to give e-mail users flexibility in attaching files to e-mail messages, and in defining new types according to their needs. E-mail clients are not expected to deal with all media types; if they cannot handle a specific data type, they just classify it as ‘unknown application.’

However, in the long-term records maintenance and archival permanent preservation processes (see section 5.5), one must guarantee the ability to render any part of a kept or preserved message at any time in the future. One should therefore make sure that:

- all media types that appear in kept or preserved messages are registered in the recordkeeping and permanent preservation systems, together with the information necessary to handle them, even if they are not registered with IANA;
- an application is available for each media type registered in the recordkeeping and/or permanent preservation systems; or
- a converted copy of the attachment is preserved as well, in a f ormat that guarantees the possibility of rendering it at a later time.

Finally, problems arise from dynamic information that may be contained in a message. Common cases include external references (e.g., Web links) and context-dependent information (e.g., date and t ime) in attached documents. Because such messages are not self-contained, there is no guarantee that they can be properly rendered at a l ater time (nor, in some cases, even at arrival time!). Therefore, when keeping and/or preserving these messages, appropriate policies should be devised, either to prevent dynamic content or to ‘freeze’ all dynamic references at capture time (or at the time of registration in the recordkeeping system).

# 4 Security and privacy issues

Security denotes the ability to manage unwanted events, by preventing them or setting up measures for mitigating consequent damage and loss. Hence, e-mail security should address the whole process in which e-mail occurs, taking into account the environment and the risk conditions.

In this section we will outline the general e-mail security aspects, identifying the main vulnerabilities and the typical risk scenario.

## 4.1 Vulnerabilities

The Internet e-mail infrastructure derives from the one originally designed for the ARPAnet, in which the only real security requirement was the capability of delivering messages even in the case of partial network failure; at the time, confidentiality, end points authentication and non -repudiation were not considered important security issues.

As a consequence, an Internet e-mail message is poorly protected against unauthorized disclosure and can easily be forged. Moreover, no mechanism is provided to detect a loss of integrity. Therefore, to make a comparison, the confidentiality of an e-mail message exchanged through the Internet may be considered comparable to that of a traditional letter mailed without an envelope.

These vulnerabilities are mostly related to the lower-level Internet protocols, mainly the TCP/IP layers, used to ship packets of information through the network. These vulnerabilities could have been handled and fixed at a higher level by e-mail protocols and formats (SMTP and MIME); however, this was not done, primarily because e-mail was mostly used within the scientific community at the time that these protocols were originally designed.

More recently, these limits have been overcome by the S/MIME standard, an extension of MIME, which supports an adequate set of cryptographic security services: authentication, message integrity, non-repudiation of origin and confidentiality. At the moment, many commercial products support S/MIME and, therefore, offer a better security level; however, interoperability problems are still frequent and, therefore, full support of S/MIME cannot be considered a standard feature.

Considering the e-mail archiving process, further vulnerabilities are related to the characteristics of the system used for storage. Nevertheless, as we shall discuss in sections 6.3 and 6.4, it is possible to at least protect the integrity of the message after it has been archived.

## 4.2 Risk scenario

Despite its high degree of vulnerability, the use of e-mail is widespread and most users are not concerned about the related security problems. The perceived risk of content disclosure or receiving forged messages is actually very low. We point out, however, that a low perception of the risk does not imply that the level of risk is actually low. For instance, we may presume that, in many environments, e-mail may be routinely scanned (at least) by intelligence offices. Indeed, unauthorized message content disclosure is very difficult to detect, and users are generally unaware of it when it happens.

On the other hand, it is worth noting that although most business, government and legal processes rely on e-mail, there is actually no evidence of significant problems arising from content disclosure or message forgery. More serious security concerns are related to other threats that do not exploit e-mail vulnerability, but take instead advantage of the vulnerability of human behavior through the use of, for example, phishing and spam.

Phishing—i.e., the process of acquiring confidential information such as usernames, passwords and credit card data—is a new and very popular form of fraud that uses e-mail as a vehicle. We shall not discuss it, since it is not relevant for the purpose of our study.

Instead spam—i.e., the huge unsolicited stream of e-mail that floods our mailboxes—needs to be carefully analyzed as an issue in e-mail archiving, since it affects the selection of the messages to be archived.

### 4.3 E-mail spam

Unsolicited messages (mostly advertisements) are frequent in all communication media. It is a sort of ‘noise’ that we have to isolate and discard to get to the actual information. The more the level of the noise increases, the more difficult it becomes to cut the noise off, and the more the communication becomes blurred.

In e-mail, spam is the noise, and it has become in recent years very intense. According to some accounts, spam volume exceeded legitimate e-mails in 2007. Even if the goal of spammers is not to block e-mail service, in reality, among the potential consequences of the huge volume of spam is some kind of denial of service.

As with every kind of noise, spam can be reduced by using appropriate (anti-spam) filters, the fine tuning of which is a very delicate task, since an improper setting may result in mistaking legitimate messages for spam. However, a sophisticated anti-spam technology has developed that is able, if properly used, to detect a significant percent of spam with a very low degree of error.

Common anti-spam products may be set according to one of the following policies:

- presumed spam messages are simply marked as spam and grouped in special folders; and
- presumed spam messages are discarded by the filter.

The choice between these polices is affected by the relevance of so-called ‘false positives’—i.e., messages that are tagged by the filter as spam but are not—and the consequent potential loss of legitimate messages.

Anti-spam filters drastically reduce the number of messages coming from known spam sources or having typical spam characteristics; however, there are other messages that may be meaningless for the recipient and the organization (e.g., jokes, unsolicited news, service messages, error messages, etc.) that still get to the mailbox.

Hence, even with anti-spam filtering, there are at least three categories of such ephemeral messages that the records management policy should consider deleting:

- unrecognized spam messages that were not blocked by the anti-spam filter;
- messages marked as spam by the anti-spam filter; and
- messages that do not have spam characteristics, but are unsolicited and are not useful for the user and/or the organization.

Filtering out also these messages is particularly important if the e-mail records management policy calls for the capture of most (all) messages using automatic capture schemes. This is still the most popular option in most commercial e-mail "archiving" products, and the policy adopted by many large organizations.

## 4.4 Message authenticity

According to the InterPARES Glossary, authenticity is "the quality of being authentic, or entitled to acceptance. As being authoritative or duly authorized, as being what it professes in origin or authorship, as being genuine."[19] For e-mail messages these characteristics should refer to the original message—i.e., the one sent by the sender's server—and encompass both the message and its metadata (for instance the subject, the sender the date, etc.). To make the point, let us look at some definitions in the RFC 2822 e-mail standard.

The `Date` header "specifies the date and time at which the creator of the message indicated that the message was complete and r eady to enter the mail delivery."[20] Namely, it is information that the sender may set up autonomously (usually the mail client sets up the `Date` field to the current client system time).

The `From` header specifies "the mailbox(es) of the person(s) or system(s) responsible for the writing of the message."[21] The standard provides also for the case where the mailbox of the author is different from the one of the person who actually sends the message ("if a secretary were to send a message for another person"[22]): in this case, the latter mailbox should be specified in the `Sender` header. Therefore, according to the standard, the client should set up the `Sender` header, while the user should set up the `From` header.

Commercial products implement mail standards with slight differences, with the aim of simplifying the user interface. A typical approach is the following:

- every header field that can be set up automatically (`Data`, `From`, `Reply-to` …), is usually set up by the client; and
- user options are provided for modifying default values, and p ossibly to set up some header values.

[19] InterPARES 2 Terminology Database, http://www.interpares.org/ip2/ip2_terminology_db.cfm.
[20] P. Resnick (ed.), Internet Message Format, RFC 2822, April 2001, p. 20. Available at http://tools.ietf.org/html/rfc2822.
[21] Ibidem, p. 21.
[22] Ibidem.

As a consequence, we tend to consider information in mail header lines as system data and, therefore, authentic insofar as the mail system is reliable. Instead, the header data should be considered user data, like the message text, and t herefore considered authentic only to the extent that we can rely on the sender and/or on the controls exercised on the process of records creation by the creator.

For instance, it is easy to forge a message and make it look as if it were coming from another person simply by setting up a nother mailbox name through the client configuration options. Moreover, in the case of forwarded e-mail, the text of the original mail may be easily modified by the new sender, compromising the forwarded message's authenticity.

Thus, an e-mail message can be considered authentic if we can assume that the sender shown in the message text and associated to the mailbox indicated in the `From` header corresponds to the actual sender. In the case of a forwarded message, we can consider the original message authentic if this condition is satisfied for all messages (original and forwarded) and we trust the forwarder(s). Of course, these are necessary but not sufficient conditions.

Regardless, these conditions cannot be easily assessed. A misleading setup of the `From` header and `Data` header may be revealed by analyzing the message header and the system data, but most users would not be able to detect this kind of fraud. Manipulation of a forwarded message may be discovered as well, but most users tend to trust its authenticity without even taking into account the possibility of text manipulation. To avoid such problems, some e-mail servers track and show user modifications when forwarding a message, but this is still an uncommon proprietary function.

Despite the ease with which e-mail messages can be forged, experience shows that e-mails exchanged in common business activities may be nearly always considered authentic. In fact, e-mails are not much more vulnerable than traditional letters and, as for paper messages, false e-mails are generally apparent when considered within their context.

In an archival process, it is more difficult to relate message authenticity to the context, or to perform crosschecks that may reveal inconsistencies. For that reason, message authenticity should be stated by the addressee before starting the recordkeeping process.

Another way to ensure authenticity is to add functionalities, based on trusted authorities, granting message authenticity. An electronic signature is an additional provision granting message authenticity. Other solutions are based on t hird party services, like the Italian Certified e-mail.

### 4.5 Certified e-mail

Certified e-mail Service (*PEC, Posta Elettronica certificata*) is an e-mail service complying with strict regulations issued by the Italian government, which enables e-mail communication that is granted by law the same value of registered mail. The service is supplied by providers, having proved technical and liability characteristics, accredited by a national body.

Certified e-mail messages can be sent among users registered with certified providers, who have to comply with security and interoperability requirements and who are supervised by a national agency. When a message is sent, in addition to the standard delivery service, the provider authenticates the sender and issues two electronically signed receipts: one proving that the message has been sent by the sender, and the other one proving that the message has been delivered to the destination mailbox. Electronic receipts have legal value and may be used in litigations. Moreover, the receipts may contain a 'fingerprint'—i.e., a digest of the content of the message signed by the certified provider—that can be used to avoid repudiation of the message content by the recipient.

Therefore, certified e-mail, thanks to the presence of a trusted third party, certifies the authenticity and the integrity of the message and provides formal evidence that the message has been delivered.

## 4.6 Privacy issues

As said before, personal data included in an Internet e-mail message may be easily disclosed, thereby exposing users to potential problems (e.g., identity theft). This issue is out of the scope of our study, but we would like anyway to point out that it is a good policy either to avoid the exchange of personal and/or sensitive data through the Internet, or to protect such data by means of encryption.

In spite of the actual nature of the threats, privacy worries concentrate more on the office environment than on the Internet. The main concern is indeed about unauthorized mailbox access, rather than about disclosure of message content during its transmission. This concern derives from making an inappropriate parallel with traditional mail, which is protected during the delivery, but may be violated at its destination when handled by the wrong person.

In fact, in countries with constitutional guarantee of the secrecy of correspondence, e-mail is equated to traditional mail, and only the owner of the mailbox is allowed to access it, even in an office environment. In these countries, privacy authorities protect e-mail confidentiality and grant the administrators the right to access users e-mail message only in particular situations and with due care.

These rules, which vary from country to country, may have a strong impact on the e-mail recordkeeping policy. For instance, in Italy, complying with privacy regulations prevails on the need to maintain information that has a potential legal relevance; on the contrary, in the United States, strict regulations call for maintaining all legally relevant information regardless of privacy issues.

In principle, employees should use company mailboxes only for business purposes and use a personal mailbox for their private messages. In practice, however, it sometimes can be very difficult to distinguish between personal and business communication; therefore, the use of institutional mailboxes is often careless. In any case, there may be business messages that, since they are meant to be read by a specific person, may also contain personal information that should not be disclosed.

Some organizations use a practical approach to cope with privacy requirements and solve the problem by asking the users to explicitly grant the organization the right to access their company mailboxes, or by providing users with a way to tag their messages as public or private, thereby allowing for a selective recordkeeping policy. However, in some countries, these procedures may not be accepted by privacy authorities.

## 5 Managing and preserving e-mail

In this section we will discuss the key components of the e-mail records management and permanent preservation processes. In doing so, we will refer to rather complete and elaborate schemes that are devised, and are realistically suitable, mainly for medium or large organizations. Of course, e-mail maintenance is an important issue also in the small office and home environment; however, we shall not discuss here this case, since the requirements are different and considerably less complex and less demanding, and simpler solutions should be envisaged.

### 5.1 Capturing e-mails

Capturing e-mails is the first, and perhaps the most delicate, phase in the e-mail maintenance process. It can basically be performed in two ways:

- *server-based capture*: incoming and out going messages are systematically captured when they get to the e-mail server, potentially after being filtered according to predefined rules; and
- *client-based capture*: incoming and outgoing messages are captured with the cooperation and consensus of the user, who interacts through the e-mail client.

Server-based capture is, in principle, the most simple and desirable option, since it allows the screening of all inbound and outbound traffic and can automatically perform the filtering of the messages to be captured according to uniform rules specifically devised to comply with a creator's organization policy. In this way, if the rules are correctly defined and consistently applied, and the system that oversees and implements those rules is reliable, no information relevant to the organization is lost.

Some level of filtering may be necessary to avoid the capture of certain ephemeral or unsolicited e-mails (e.g., spam). This can be performed with rules based on mailbox specification and on message content and metadata, such as the sender, the recipient(s), the subject and the dates sent and received. Automated classification tools can also be integrated in the filtering system to perform preliminary tagging of captured e-mails.

However, as we have discussed in section 4.6, in some countries, legal ownership of e-mail is unclear, and privacy regulations may prevent the automatic capture of e-mails. In extreme cases, even recording the arrival or the departure of a personal message may be considered a violation of the individual's privacy.

A simple solution to this problem, adopted by many organizations, is to inform users that all messages going through their mailboxes that comply with certain filtering rules are going to be captured, and to ask them to use 'personal' mailboxes, outside of the capture mechanism, for their private e-mails.

In other cases, asking the user's consensus for every specific message capture may be necessary, and a client-based message capture scheme will have to be implemented.

However, pure client-based capture has several drawbacks, since message filtering relies on the decision of individual users, who may fail to apply correctly and uniformly the organization's message filtering criteria.

Server-based message capture is extensively used by corporations and other large organizations, because it can be performed automatically, without putting any burden on the user, and it overcomes privacy and confidentiality issues (if appropriate access schemes are implemented). Moreover, it has the further advantage of saving the messages as soon as they arrive on the server.

In these cases automated message capture schemes are generally appropriate, since the final purpose is not to file e-mails in a r ecordkeeping system, but simply to temporarily store them, mostly too meet legal compliancy requirements.

Instead, in a more general scenario, the making and keeping of e-mail records is likely to require the intervention of the user (i.e., utilize client-based message capture), both because of privacy and confidentiality and because of the need to determine when an ingoing or outgoing message needs or deserves to be declared as a record (i.e., classified and registered) and filed into the recordkeeping system.

A 'mixed approach' that capitalizes on the unique advantages of both server-based and client-based capture schemes is the following:

- a first level message filtering is performed at the server level, weeding out certain ephemeral and non-relevant messages;
- candidate messages are "proposed" to the user using automated filtering and records classification routines built into the system, and the user is prompted by the system for consensus on the classifications applied by the system and/or on whether the messages should be declar ed as records and be filed to the recordkeeping system; and
- individual users retain the capability of independently capturing and/or declaring as a transitory or corporate record any message that they send or receive.

Regardless of the filtering scheme adopted to decide if a m essage is going to be captured and declared a r ecord, the user may, and shall probably, be involved in the classification of the records and in manually entering additional metadata.

## 5.2 Recordkeeping and preservation formats

As for any digital record, the ongoing maintenance and preservation of an e -mail message must ensure two conditions:

- the original structure and a ll the information contained in the message must be retained; and
- future users must be able to access the information in the message in its original documentary form—i.e., manifested to future users in the same way that it was manifested to the original users (sender and recipients).

This means that not only the content, but also the structure and the appearance of the message must be preserved.

As we have discussed in section 2.7, the standard e-mail format is the one defined by RFC 2822, which is limited to messages in plain ASCII*,* with the subsequent MIME extension (RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289) to support text and headers in character sets other than plain ASCII, messages structured in multiple parts and non-text attachments.

As we already pointed out, although not yet official Internet standards, RFC 2822 and MIME, which are currently *draft standards*, are universally used and acknowledged by everyone. A message in this format, including all its metadata and attachments, is a single plain ASCII file, which means an object of very simple structure that is very easy to store and maintain or preserve.

Therefore, the RFC 2822/MIME format should always be the primary maintenance or permanent preservation data format for e-mail messages. Moreover, this solution is easy to implement, since this is the format used by most e-mail servers and clients to store messages internally.

The RFC 2822/MIME format guarantees that all the information is retained, and the structural integrity is maintained, but the rendering of the information in its original form is guaranteed only for messages created in plain ASCII, which are today a small minority of all messages. Instead, messages exploiting the full MIME format, i.e. with attachments in a variety of MIME media types rely on external applications to be decoded, reconstituted and manifested to the user.

A future user can therefore access an attachment in the MIME encoded form, but may be unable to actually access its content, unless the corresponding application is available. This is indeed a well known problem in digital records preservation, since all digital records rely on an appropriate hardware-software environment to be correctly rendered.

As a principle, to ensure future access to the records, one should maintain the original hardware-software environment, or, at least, for every MIME media type registered in the recordkeeping or permanent preservation systems, maintain software applications having certified compatibility with the original ones. Because of the rapid rate of technical obsolescence, this is, of course, no easy task, especially over the long term.

Actually, to carefully assess the relevance of this problem we must make clear distinction between two different kinds of scenarios:

- *short-term maintenance*, when messages must be maintained and accessed for a short period of time, typically up to ten years; and
- *long-term maintenance*, when messages must be maintained and accessed for a long period of time, typically more that ten years.

At the moment, the large majority of organizations are mostly interested in the first scenario, primarily because of the need to meet regulatory compliance requirements, but also because most commercial "e-mail archiving" products, which we shall discuss in more detail in section 7.3, are designed to meet these short-term maintenance needs. We shall therefore discuss in this section only the short-term scenario. The long-term maintenance scenario, which deserves a different and more complex approach, will be discussed in section 5.6.

In the short-term scenario, access to maintained messages with attachments in a variety of MIME media types does not pose special problems and can be granted rather easily, by means of a few very simple provisions. In fact, we may conveniently assume that, in cases where a message is registered in a recordkeeping system by an organization shortly after it has been sent or received, the current hardware-software environment in the organization will allow the user who has sent or received the message to read it, with all its attachments. In the short term, the same kind of access can therefore be granted also to all 'recently' registered messages, directly through the e-mail client interface.

What must be done to ensure this is simply to make sure that the software applications for MIME media types in all currently kept messages are maintained, as well as the hardware-software platform needed to run them.

Moreover, to conveniently support presentation and search and discovery actions (see sect. 6.1), which may include searching by content, it can also be useful to store copies of 'printable' attachments, converted in standard searchable print-image format (e.g. PDF), as separate records linked to the message.[23].

Summarizing, in the short-term maintenance scenario:

- messages are maintained in RFC 2822/MIME format to help ensure their continued authenticity;
- attachments are extracted as binary files, and st ored in the recordkeeping system as separate records, linked to the main record;
- 'printable' attachments are also optionally converted to a print-image format and kept as separate records, linked to the main record, to support search and discovery actions;
- a database of MIME media types in all currently maintained messages and the indication of all corresponding software applications are maintained; and
- actions are taken to guarantee the availability within the organization of all the software applications listed in the database and of the hardware-software platforms needed to run them.

## 5.3 Message classification and metadata extraction

There are basically two options in implementing message classification, which may be considered independently or in combination:

- messages are classified by means of automatic classification tools; and/or
- users (i.e., senders or recipients of messages) are requested to provide a classification code based on an established classification scheme or naming convention.

[23] As is suggested in section 5.5 for long-term maintenance and/or preservation scenarios, attachments that are not 'printable' (e.g., audio, video, etc.) should be converted to the most suitable supported standardized file format, maintained as separate records and linked to the main record. Whether this same approach is necessary or appropriate for these types of attachments under the short-term recordkeeping scenario will vary depending on the requirements of the users and resources of the organization.

The first option is typically used in a server-based capture scenario (see section 5.1), since the filtering process naturally provides some degree of classification, based on the rules that have been used to choose to retain the message. Simple automated e-mail classification tools classify sent and received messages based on main message header metadata, but some sophisticated tools may exploit also message content (i.e. information in the message body and the attachments).

Automatic classification is a common strategy used by large corporations: (1) to deal with huge volumes of messages, and/or (2) to deal with capture schemes that do not, or cannot, involve final users.

However, at the state of the art, automatic classification procedures have an insufficient level of reliability, and therefore, when considering the recordkeeping level, automatic classification could be inappropriate, at least in more demanding environments as government agencies, or in institutions that have explicit legal recordkeeping responsibilities. On the other hand, giving the users full responsibility may put too much of a burden on them.

As for the message capture, a mixed approach could be the appropriate option. The server-side system, using an automated classification tool would propose a set of possible classifications, derived from the message content and/or metadata, and the user would only be requested to make the final choice, or, if necessary, to override system proposals by introducing his/her own.

Classification schemes depend on the unique needs and requirements of individual organizations, and are in general not specific to e-mail records; therefore we shall not discuss here classification metadata, for which one should refer to the general literature. We shall instead discuss in this section metadata that are specific to the peculiar nature of the e-mail message, which we shall call *message metadata*, and we shall refer in doing that to the short-term maintenance scenario discussed in sect. 5.2.

As we have seen in section 3.2, valuable information about message origin, destination and delivery is contained in the message header. Hence, certain metadata can be directly extracted from the header lines. More precisely, all the items in Table 1 in section 3.2 that are marked with A (for *always*), are always present in the header; hence, these can be taken as *mandatory metadata*. The remaining items, marked with F (for *frequent*) or with O (for *optional*), can be taken as *optional metadata*.

In addition to these, further metadata are often added to keep track of the delivery of outgoing messages. These can be automatically extracted from delivery reports that can be linked to these messages by means of their message identifiers.

Metadata should be provided also for each of the message attachments, if any, and should specify:

- the MIME media type;
- the computer file name;
- the application that should be used to open the computer file; and
- the link to the attachment, if it is stored in the recordkeeping system as a separate record.

Further mandatory metadata are used to improve the specification of the message sender and recipient(s) by recording their ‘intelligent names’ (i.e., the ‘real life’ names associated with their e-mail addresses). Intelligent names do not necessarily exist and are not necessarily in the message header. Eventually, this information may be extracted, during the message capture, from the address book of the user who is the sender or the recipient of the message.

Another important issue is the use of mailing lists in the recipient fields. In the case of a server-managed list, the address field in the header only contains the name of the list, and this is often done on purpose so that each recipient is not able to see in his/her copy of the message the addresses and the names of the other recipients.

On the sender side, this problem can be handled at capture time, either by accessing the list on the server and generating a separate message header for every recipient in the list, or by maintaining the lists as separate records and referring to them in the message header. In the latter case, as the content of mailing lists evolves, a complete set of each list version should be maintained.

Mandatory and optional *message* headers are shown in Table 2. These, as we already pointed out, are only headers feeding metadata specific to the peculiar nature of e-mail messages; for other metadata, one should refer to the general literature (see appendix A). A rather large set is presented in the table, and an asterisk in the first column is used to specify what should be considered as a complete minimal set.

## 5.4 Checking and maintaining authenticity

As we have seen in section 4.4, assessing the authenticity of an e-mail message is a nontrivial task. Because the e-mail infrastructure and, chiefly, the Internet through which messages are transmitted, are, largely, unprotected, e-mail records are potentially exposed to unauthorized manipulation.

The potential for unauthorized manipulation is, of course, also a concern for traditional paper records. In general, however, moving from the traditional paper environment to the digital environment, does not improve the situation—and, in fact, exacerbates the situation in most cases—and does not, therefore, decrease the need to care about record authenticity.

In contexts characterized by high risk of forgery, when classifying and registering an e-mail message in the recordkeeping system, we can collect and attach as metadata to the record all the information that has been used to check its identity[24] and its integrity;[25]

[24] Identity is defined by InterPARES as “The whole of the characteristics of a document or a record that uniquely identify it and distinguish it from any other document or record. With integrity, a component of authenticity.”

[25] Integrity relates to the quality of a record being complete and unaltered. Because in the digital environment there are no "original" records--only record copies that are generated each time the record's stored digital components are reconstituted and manifested in the proper documentary form for the user--integrity is always adjudged in relation to the first manifestation of the record. Strictly speaking, for a digital record, this means that the stored digital components corresponding to the first manifestation of the record are maintained, and not a single bit is changed. However, in some contexts, the definition of integrity may be slightly more flexible, only requiring that the essential parts of the record (i.e., those intrinsic and extrinsic elements of documentary form that are required to ensure that the record is a complete and effective record--that is, capable ) remain unaltered, or else are only altered within acceptable limits such that the record remains *effective* (i.e., capable of reaching the consequences or producing the effects for which it is intended).

**Table 2.** Message metadata

| | NAME | DESCRIPTION | SOURCE | OBLIGATION | OCCURS |
|---|---|---|---|---|---|
| * | Message-ID: | *Unique message identifier* | `Message-ID` header | optional | once |
| * | Message-type | *Outbound or inbound message* | SMTP server | mandatory | once |
| | Message-reference | *ID of message referenced* | `References` header | optional | once |
| | Reply-to | *ID of the replied message* | `In-Reply-To` header | optional | once |
| * | Date-sent | *Date and time the message was sent* | `Date` header | mandatory | once |
| * | Date-received: | *Date and time the message was received (for inbound messages)* | Date and time in last `Received` header line in message header | mandatory for inbound messages | once |
| * | Date-captured | *Date and time the message was captured* | E-mail archiving system | mandatory | once |
| * | Subject | *Subject of the message* | `Subject` header; user | optional | once |
| | Description | *Description of the content* | `Comment` header; user | optional | once |
| | Keyword | *Keywords* | `Keyword` header; user | optional | many |
| * | Author | *Message author* | `From` header | mandatory | once |
| | Sender | *Message sender (on behalf of author)* | `Sender` header | optional | once |
| * | Author-name | *Intelligent name of author* | `From` header | optional | many |
| | Sender-name | *Intelligent name of sender* | `Sender` header | optional | many |
| | Organization | *Organization of the author/sender* | `Organization` header; user | optional | once |
| | Re-sender | *Sender forwarding the message* | `Resent-From` header | optional | once |
| * | Recipient-address | *E-mail address of recipients* | `To` header | mandatory | many |
| * | cc-Recipient-address | *E-mail address of cc recipients* | `cc` header | optional | many |
| * | bcc-Recipient-address | *E-mail address of bcc recipients* | `bcc` header | optional | many |
| * | Recipient-name | *Intelligent names of recipients* | `To` header | optional | many |
| * | cc-Recipient-name | *Intelligent names of cc recipients* | `cc` header | optional | many |
| * | bcc-Recipient-name | *Intelligent names of bcc recipients* | `bcc` header | optional | many |
| | Resent-recipient | *Address of recipient of the resent message* | `Resent-to` header | optional | many |
| * | Structure | *Upper level MIME content-type* | `Content-type` header in message header | mandatory | once |
| * | Attachments | *Number of attachments* | Message structure | mandatory | once |
| * | Attachment-ID | *Internal attachment ID number* | ERMS | optional | many |
| * | Attachment-type | *Media type of the attachment* | `Content type` header in message parts | optional | many |
| * | Attachment-name | *Filename of the attachment* | `Filename` header in message part | optional | many |
| | Attachment-link | *Link to attachment record* | ERMS | optional | many |
| | Notification | *The message requests disposition notification* | `Disposition-notification` header | optional | once |
| | Notification-sent | *Whether notification was sent* | SMTP server log | optional | once |
| | Notification-received | *Link to notification message* | SMTP server log | optional | once |

that is, its authenticity. Such information can then be used by future users to assess the trustworthiness of the e-mail record.

For incoming messages, besides all the information in the message header (see section 5.4), capturing and maintaining the message in the RFC 2822 format ensures that all the data about the sender, the transmission path and t he dates are saved and protected. Moreover, as the message is saved in its original format, exactly as it was delivered to the receiving server, this is an essential element for any future control.

For outgoing messages, the e-mail server log files, which, along with the bounce messages and the thread metadata, should be retained as well, help to assess when the message was actually sent, and if and when it was delivered to its recipient(s).

A stronger assessment on the identity of the sender and the integrity of the transmission can be m ade through the use of an electronic signature, which is becoming widely adopted, especially when e-mail messages are used to transfer documents with legal value or records of business transactions.

We may distinguish two cases:

- the message is electronically signed by the sender, either using S/MIME format or by applying another standard signature (e.g., XML signature) to the attachments; or
- the original message is not electronically signed by its sender.

In the former case, some organizations require that the message be r etained and maintained in the original signed format, together with information needed (such as the X.509 electronic certificate[26]) to verify the signature. According to MoReq2, an ERMS platform should encompass the following features for capturing and m aintaining messages bearing an electronic signature:

- electronic signatures, associated electronic certificates and details of related certification service providers, should be archived as separate records and linked to the record to which they pertain;
- certificates should be checked against the revocation lists to assess their validity at the time the message was captured; and
- signed files should be verified, and t he details and outcome of the verification process should be stored as message metadata.[27]

If the original message was not electronically signed by the sender, some organizations require that it be electronically signed at the time of capture and/or registration in the recordkeeping system, to ensure that at least some aspects of the identity and the integrity of the message are maintained during the transient storage and/or subsequent recordkeeping phases.

[26] See http://www.itu.int/rec/T-REC-X.509/en.

[27] Serco Consulting, *Model Requirements for the Management of Electronic Records: Update and Extension, 2008. MoReq2 Specification, v1.04* (Bruxelles, Luxembourg: European Commission, Office for Official Publications of the European Communities, 2008), pp. 143-146. Available at http://www.cornwell.co.uk/moreq2/MoReq2_body_v1_04.pdf.

When dealing with large numbers of messages, some consider an effective way to augment the implementation of secure e-mail record capture and retention procedures is to group messages into batches and sign each batch as a single file, retaining the electronic signatures and the certificates as separate records linked to the corresponding batches of messages.

Encryption is used in e-mail to protect the confidentiality of the content during message transmission, and may be performed either by the e-mail system or by the user. System encryption often occurs at the front-end level, via VPN (Virtual Private Network). In this case, both the user and the mail server are unaware of the cryptographic process and the recordkeeping activities are not influenced by the encryption process.

As said before, encryption may also be performed at the e-mail-client or the e-mail-server levels according to the S/MIME standard; however, interoperability problems still hamper the diffusion of this kind of protection. Therefore, quite often, encryption and decryption are performed by end users by means of cryptographic functions of commercial products (e.g., cryptographic options of Microsoft Word, Adobe Acrobat, PGP). In general, decryption of such messages requires the cooperation of the user who is the message recipient and, presumably, the owner of the key.

Ideally, messages should always be captured and maintained in the form in which they were intended to be manifested; therefore, if transmitted in encrypted form, they should be decrypted before being saved and filed in recordkeeping system.

Achieving this may be problematic in case of automatic data capture for e-mail retention, since the capture of an incoming message has to be performed before the recipient may handle the message, because of regulatory compliance and legal discovery requirements. Therefore, in juridical-administrative contexts where these requirements are stringent, the best policy is usually to ban user encryption and provide for system level encryption.

A different approach is sometimes followed when registering a message in a recordkeeping system, provided the user could be involved in such process. In this case, when registering in the recordkeeping system encrypted messages, or messages with encrypted attachments, the following provisions are taken:

- the message and/or its attachments are registered as separate records both in encrypted and decrypted form;
- metadata are added to each record with all encryption details, such as the encryption algorithm, the decryption key (or the electronic certificate, when applicable) and the level of encryption used; and
- metadata are added to each record with decryption process details, such as date and time, decryption software used and the name of the person responsible for the decryption process.

Maintaining the records in the decrypted form is especially important for the long-term maintenance scenario (see section 5.5), since encryption is likely to reduce the ability to access the records in the long term due to the potential unavailability of the required decryption software and/or to the loss of the decryption key.

Another practice used to protect the integrity of an e-mail record—both the whole message and its attachments, if stored separately—is to generate *digests* (i.e., digital fingerprints) for all of these objects. These should be kept separately, linked to the corresponding records, and possibly electronically signed by a trusted records officer.

Besides all the message capture and recordkeeping actions and provisions that we have discussed in this section, the maintenance of the authenticity of the kept records strictly depends, as we shall discuss in the next section, on the access control schemes used in regulating user access to the records and on the audit procedures deployed to guarantee accountability for all of these records management activities.

## 5.5 Long-term maintenance

As we pointed out in section 5.2, most organizations are only interested in short-term maintenance of e-mail messages (i.e., with a time horizon up to ten years). Long-term maintenance, at least as far as e-mail is concerned, is a problem that concerns only a limited number of large organizations. In these cases, e-mail messages are managed together with many other types of digital records, and their long-term maintenance may benefit from large scale factors and the support of an efficient and complete trusted recordkeeping or trusted records preservation infrastructure.

Long-term maintenance poses two kinds of problems:

- ensuring the authenticity of messages over the long term; and
- ensuring the ability to provide continued access to all the information contained in the messages and in their attachments.

The first problem is a general one in long-term digital information maintenance and preservation, and really there is nothing specific to the e-mail case. As with all digital objects in general, it is a matter of saving the digital components of the records in non-volatile storage on reliable digital media; controlling the technical obsolescence of both the hardware and the software required to reconstitute the digital components and manifest them to users as records in the proper documentary form; controlling the technical obsolescence of the records themselves as the hardware-software environments in which the data or file formats of the records operate evolves (e.g., through transformative migration[28] of the records to maintain their accessibility in updated technological regimes); and monitoring the integrity of both the stored digital components and the media on which they are stored to decide when new copies of the records should be produced (i.e., refreshing of records), eventually with new technologies. Therefore, we will not discuss this matter here, and one should instead refer to the general long-term records maintenance and permanent preservation literature.

The second problem, as we have seen, deals with MIME media types and the long-term maintenance of hardware-software environments necessary to handle them, and has some specific aspects in the e-mail case:

[28] Defined by InterPARES as "The process of converting or upgrading digital objects or systems to a newer generation of hardware and/or software computer technology."

- the variety of MIME media types and subtypes used in the creation of digital documents in general is extremely large; and
- there is a general lack of control over the document creation process in most e-mail environments: in some cases, e-mail users may include attachments in any registered and supported MIME media type, while in some other environments organizations are able to strongly recommend, or even enforce, the use of data formats more suitable for long-term maintenance.

The approach of maintaining the applications and the hardware-software environment needed to run them that we have discussed in section 5.2 for the short-term maintenance scenario may realistically be out of question for the long-term maintenance scenario, unless we wish to transform National Archives into ICT museums.

Pragmatically, the only solution considered reasonable is to convert the messages and all their attachments, *preferably as soon as they enter the recordkeeping system* into standardized data or file formats that are realistically possible to support over the long term.

More precisely:

- messages should be maintained in RFC 2822/MIME format to help preserve their authenticity; in a future time, if applications are still available, the attachments could still be accessed in their original format;
- attachments that are 'printable' should be converted to a supported standardized print-image format, maintained as separate records and linked to the main record;
- attachments that are 'not printable' (e.g., audio, video, etc.) should be converted to the most suitable supported standardized format, maintained as separate records and linked to the main record;
- a database of all converted records and their data or file formats should be maintained;
- when a supported data or file format approaches obsolescence, all records in that format should be converted to a new 'equivalent' supported format; and
- information about the original data or file format and the details of all conversion processes to which the records have been subjected should be registered as message metadata for all converted records (or for their individual digital components, if relevant); this provides some kind of assessment of the conversion procedure, and allows future users to assess to what extent the integrity of the record may have been compromised.

As a final remark, we shall point out that, when messages are preserved for historical purposes, the main goal is usually to preserve the *integrity of the information in the message* at a semantic and semiotic level, even if the *integrity of the message* is "compromised" by a format conversion that introduces slight changes in the rendering of the record's documentary form.

A future user, reading in 2050 the converted copy in PDF/A v. 47.1 of an attachment originally created in MS Word 2003 .doc file format, may get all the information s/he needs, and be comforted about the trustworthiness of this information by the assessment of the archivist who, in 2031, performed the last conversion. Anyway, if he is not satisfied with this, the alternative for him could just be the contemplation of a binary file.

# 6 Access to stored e-mail records

For an organization to grant users with efficient access to e-mail records, while still protecting the records from unauthorized access and accidental or fraudulent manipulation or destruction, it must address several issues:

- provide search and discovery capabilities, which should be powerful and flexible because of the amount of records and because search criteria are not known in advance and can be very variable and unpredictable;
- provide adequate presentation capabilities—i.e., the user should be able to efficiently view and examine the content of the messages retrieved by a search or discovery action, including all their attachments;
- define access control policies and access privileges and restrictions to protect message integrity, and to avoid unauthorized access to protected information;
- enforce access privileges and restrictions; and
- routinely audit the effectiveness of access control policies and procedures; and automatically generate audit logs, for accountability and, possibly, for data recovery purposes.

## 6.1 Search and discovery

As we already pointed out, search and discovery capabilities are among the main reasons that induce organizations to deploy e-mail recordkeeping systems.

Search is based both on message content (text and attachments) and metadata, and the use of relational and Boolean operators are allowed to combine an unlimited number of search terms. Moreover, the system allows the use of propositional search logic, with partial matches and wildcard characters. Proximity search is also an important feature, to find terms separated by no more than a specified number of words.

Search and discovery capabilities are considerably extended through the use of thesauri and ontologies to enable the user to search by concept instead of searching by specific lexical terms. This allows retrieval of records with a broader, narrower, or related term in their content or headers. For instance, a search for “transportation” may retrieve “car” or “train.”

The number of e-mails records in the system, and therefore the number of records retrieved by a search operation, may be very large. It is therefore important to allow the user to effectively limit the scope of the search, either by restricting the search to a specific portion of the records, or through an incremental search—i.e., performing at each step a further search in the results of the previous step.

Finally, the screening of search results is usually improved by ranking them in order of relevance, as is currently done by Web search engines that use a variety of sophisticated ranking algorithms to accomplish this task. Such algorithms could successfully be adapted to the peculiar structure of e-mail messages, and to the pattern of their metadata.

## 6.2 Presentation

The result of a search, or in general any record accessed by the user, are presented by the system to be vie wed and analyzed in all its components. This should be done without the need of any additional software application.

The recordkeeping system is supposed to provide viewing mechanisms capable of displaying the messages and their attachments, at least for all frequently used file and data formats, even though the generating application is not present. This poses a series of practical problems, because of the wide variety of MIME media types and the consequent burden to incorporate in the system all the corresponding applications, and to maintain them.

However, we must carefully distinguish between two different needs:

- allowing easy inspection of the content of the messages and their attachments during the search and discovery process, which may consist of several steps and require the scrutiny of intermediate results; and
- maintaining the integrity of the messages.

A practical solution to the first requirement is attained by saving converted copies of the attachments, possibly in standard print-image formats, as we already discussed in section 5.5 for the long-term maintenance scenario. This drastically reduces the number of viewing applications that have to be incorporated in the system.

On the other hand, the integrity of the message is not violated, since the converted images are only additional records, for access convenience, and the message is still preserved in its original RFC 2822/MIME format.

In the short-term scenario, a m ost common solution is to allow retrieval of filed e-mail records directly from the e-mail application. This capability is indeed supported by most commercial "e-mail archiving systems", which integrates the new functionalities in the standard familiar e-mail client interface.

## 6.3 Access control

As is emphasized in MoReq2, "it is essential that organizations are able to control who is permitted to access records and in what circumstances, as records may contain personal, commercial or operationally sensitive data."[29] This is especially true for e-mail, because of the privacy and confidentiality issues that we already discussed.

[29] Serco Consulting, op. cit., p. 40.

Access control is typically achieved by the specification and implementation of security policies—i.e., access privileges[30] and restrictions[31] are applied based on the role an individual plays in the organization. To make access management more efficient, groups and roles are usually defined, so that privileges and restrictions may be also applied, with a single action, to a group or a role, and consequently inherited by all the users belonging to that group or playing that role.

Access rights should be taken into account not only as far as the direct and explicit access of records is concerned, but also when an user performs search and discovery actions, to prevent indirect access and inference, and the consequent disclosure of information that user is not allowed to access. Consequently, no search or retrieval function must ever reveal to a user any information (metadata or record content) when the user's assigned access privileges and/or restrictions do not authorize the user to have read access to the records in question.

Consistently, when a user performs a content or metadata search, the system should not include in the result any record for which the user does not have read-level authority. The mere act of revealing that such records exist, or even how many records that fit in a given criterion are in the system, may result in the disclosure of sensitive or confidential information.

## 6.4 Audit log

As is the case with all records, accountability for the actions applied to e-mail records, is necessary to guarantee the integrity of the records, and the continuity of the chain of custody that may be used to prove it.

Detailed information about all user accesses should automatically be recorded by the system in an *audit log*, to show whether business rules are being followed and to ensure that unauthorized activity can be identified and traced.

In particular, the system should be capable to record information in the audit log for the following events:

- capture, declaration and/or registration of a message;
- definition of message header lines or metadata;
- change of message header lines or metadata;
- access to a message;
- relocation of a message within the system;
- export of a message to another system;
- disposal of a message;
- registration of a new user in the system;
- change of access privileges and/or restrictions of a user; and

[30] Access privileges are defined by InterPARES as "The authority to access a system to compile, classify, register, retrieve, annotate, read, transfer or destroy records, granted to a person, position or office within an organization or agency."

[31] Access restrictions are defined by InterPARES as "The authority to read a record, granted to a person, position or office within an organization or agency."

- cancellation of a user from the system.

For each of these events the system records in the audit log the following information:

- the type of action carried out;
- the date and time;
- the users involved in the action and their roles;
- the object involved in the action; and
- all other information necessary to reconstruct the state of the system before and after the action.

To serve its purpose, the audit log has to be unalterable—i.e., it must be impossible for any user, including administrators, to change or delete any part of it. The level of assurance needed will depend on the organization and on the level of security of the underlying operating system and system software.

A rich set of requirements for search and discovery, access control and audit are given in MoReq2, in DoD 5012.02-STD, in *Requirements for Electronic Records Management Systems* of the UK Public Record Office and in other similar documents (see Appendix A).

## 7 Commercial products for e-mail management

E-mail usage is so extensive that basic e-mail functionalities are usually included in current operating systems or browsers. For instance the Windows Vista operating system provides for the Windows Mail client to offer basic e-mail functionalities. Nevertheless, many other products are available, which extend the basic e-mail functions and provide a wide range of additional functionalities.

In the following sections we will discuss some of these functionalities, and show how these functionalities may be exploited in the various message creation, capture, storage, use, maintenance and disposition (i.e., management) processes. However, before going into a detailed discussion of commercial products, we shall first point out that these products may be aimed at two different kinds of user environments: organizations and individual users.

So far in our discussion we have mainly referred to e-mail management in a medium or large organization. But, as we already pointed out in section 5, e-mail management is also an interesting issue for small organizations, independent professionals and private users, a market segment usually referred to as *soho* (small office and home). In these cases, even if the motivations may be the same as in large organizations, the solutions may be different, and may be strictly influenced by the characteristics of specific products.

Hence, in the following discussion we shall consider how commercial products fit the requirements of both large organizations and small offices and individual users. We would also like to point out that specific commercial products are discussed with the only purpose of providing an example of typical product profiles. This survey is therefore far from exhaustive, and we recommend that any product selection for actual implementation purposes should be preceded by a thorough analysis of the current market situation.

## 7.1 E-mail clients

E-mail clients offer several functionalities that may be exploited in the various e-mail management processes, both in the case of large organizations and in the case of small offices and private users. In this section we deal with the latter case: that is, we show how e-mail management processes can be performed by individual users and small office organizations just by using e-mail client functionalities. The case of large organizations, which mostly rely on different software products, will be discussed in the next sections.

Altogether, there is a large variety of commercial e-mail clients, both proprietary and open source. These products are mainly designed to operate on PCs connected to the Internet, and therefore they offer a variety of functions to integrate e-mail communication with other forms of network communication and collaboration. Typical examples are directory services, address book and calendar management, news notification, event notification and content filtering. Major e-mail clients also support important security functions like authentication, encryption, electronic signature, certificate and revocation list management.

Since these products are designed to operate in an open environment, the functions they support are mostly based on acknowledged standards, either consolidated or emerging. As for message capture, storage, use and maintenance processes, the main features in an e-mail client to be considered are:

- ability of fetching selectively messages from the mail server, according to user-defined rules;
- possibility of labelling messages to state their relevance in transitory, short-term and long-term maintenance processes;
- possibility of adding metadata to messages for classification purpose;
- capability of grouping messages in virtual folders defined by the user;
- availability of filters for spam messages or other ephemeral messages;
- availability of backup functions;
- availability of “wizards” for assisting the user in maintenance activities; and
- support of standard formats for recorded messages.

All clients support the POP3 protocol for fetching messages from the SMTP server, and many support the IMPAP protocol as well (see section 2.6), but only a few of them allow users to select messages of interest by marking them or defining suitable rules (e.g., Alpine, Opera mail, Pegasus).

All products allow, in different ways, users to mark messages to be kept by tagging them with colors or flags. Some products allow marking messages as “not to be deleted” to protect them against accidental removal (e.g., Gnus, Pegasus, Zimbra). Unfortunately, no product provides for automated retention functions based on predefined marks. The selection of records to keep on the basis of flag/color has to be performed manually or by means of user-defined filters.

Classification metadata are currently not supported by commercial e-mail clients. Some products allow for adding notes (e.g., Apple mail, Outlook, Pegasus), but none provides for the definition of classification fields and for automatic classification. Classification must still be manually performed by labeling messages or by moving them into folders, consistently with the classification scheme.

User-defined virtual folders are supported by most commercial products, with a few relevant exceptions (e.g., Outlook express), and give the user the ability to arrange messages according to specific criteria, thus facilitating the classification task.

Filtering of unwanted or dangerous messages (spam, phishing, malware) is usually performed at the server level, or by the Internet provider, by means of specialized products. Nevertheless, many e-mail clients offer filtering capabilities (e.g., Gnus, Eudora, Kmail, Mozilla Thunderbird, Outlook, Pegasus, Pine). Generally, filtering functions in e-mail clients are very rudimentary, especially as far as the setting of the notification policy and the tuning of the filtering action are concerned.

Backup functions are meant to recover data in case of failure and, therefore, cannot be properly considered as maintenance (or preservation) features. Anyway, at least as far as private users are concerned, backup files may be conveniently stored by the user with the aim of maintaining related information. Almost all commercial products offer backup functions, and some of them (e.g., Lotus notes, Outlook) provide also scheduling capabilities.

No current product provides functions (so called ‘wizards’) to help the user in systematically setting up and c arrying on message maintenance. Therefore, users interested in maintaining messages, whether in short-term or long term maintenance phases, have to set-up their own procedures, which may be based on two alternatives:

- converting individual messages into text files and maintaining these files; and
- performing regular e-mail backups and maintaining them.

In the latter case, since backups may have proprietary format, the user should inquire about the backup format and assess its interoperability with other applications and systems, to make sure that backups can be accessed in the future also from different e-mail products.

Single messages are usually stored in the original RFC 2822 format. Collections of e-mail messages are stored by some products in proprietary formats (e.g., Lotus notes, Outlook, Pegasus), and by other products in open formats (e.g., Alpine, Gnus, Eudora, Kmail, Mozilla Thunderbird, Novell evolution, Opera mail). Products following the latter approach often give the user choice between several different open formats.

Popular formats for aggregations of e-mails are `Mbox` and `Maildir`. In `Mbox`, all messages are concatenated and stored as plain text in a sin gle file, while `Maildir` uses a separate file for each message. A significant advantage of these formats is that, since they rely on standard files, the stored information may also be accessed using standard content management tools.

As most of the e-mail records management actions have to be performed manually by the users, some e-mail products offer the possibility of automating a se quence of actions, by supporting scripts (e.g., Lotus notes, Mozilla Thunderbird, Outlook express) or java language (e.g., Lotus notes).

## 7.2 Integrated systems

As we have seen in section 2, e-mail clients must connect to an e-mail server to send and receive messages. To do so, individuals and small organizations usually rely on a service supplied by the Internet provider, which actually manages the e-mail server, and may also offer some basic e-mail records management services, but, in general, not a very complete set of services.

Medium and large organizations may instead find it more convenient to manage directly, through their ICT department, an e-mail server connected to the organization's intranet. In this case, even if in principle the e-mail server and the clients could be chosen regardless of the organization's IT context, a very popular approach is to manage the e-mail through integrated systems, according to the schema discussed in section 2.2 and depicted in Figure 3. As already noted, this market is currently dominated by two products: Microsoft Exchange Server and IBM Lotus Domino.

Exchange Server (currently the 2007 version) is Microsoft's solution for communication and collaboration within enterprises. This product is fully integrated with all Microsoft products for enterprise automation (the Server 2007 family) and with Microsoft e-mail clients, notably Microsoft Outlook. In this proprietary environment, the client-server communication, instead of using the standard open protocols discussed in section 2.6., is based on a Microsoft proprietary protocol called MAPI (*Messaging Application Programming Interface*), which is supported also by some non-Microsoft clients (e.g., Lotus notes, Zimbra).

Anyway, following a widespread and positive trend towards non-proprietary solutions, recent versions of Exchange Server also support standard access protocols (POP3 and IMAP4). Moreover, version 2007 is characterized by a high level of integration with a large variety of enterprise communication media, including instant messaging and telephone.

Microsoft Exchange stores the information contained in an e-mail message in two different ways: the message stream—i.e., the message in the native RFC 2822 format, which is saved in an `stm` file—and the so called MAPI information, i.e. message header plus proprietary information, which is stored in a database in the `edb` proprietary database format. The MAPI database allows for optimized message retrieval, but may be accessed only by clients supporting this proprietary format (e.g., Office Outlook).

Exchange Server 2007 allows for the definition and implementation of basic e-mail records management policies, through the *Exchange Management Console* and the *Exchange Management Shell*. This allows:

- controlling content retention and removing content that is no longer needed; and
- journaling (copying) important content to a separate storage location outside the mailbox.

The latter feature may also include message classification, which is performed by attaching to the item a user-selected classification label. How the classification is carried out depends on the client. For instance, if Outlook 2007 is used, the administrator may define a message classification scheme, and the client would prompt the user to choose among the available classification options.

Exchange is also tightly integrated with *SharePoint*, the Microsoft document management and co ntent management platform. In particular, *Microsoft Office SharePoint Server 2007* and *Microsoft Exchange Server 2007* allow the implementation of an integrated system that supports document lifecycle management from creation through disposition, according to specific records management policies.

*Lotus Domino* is the IBM product for enterprise e-mail management, and i s the other major player in this market. Indeed, Lotus Domino is a platform that provides also many other services, like Collaboration server, Application server, Web server, Data base server, Directory server, and more.

Lotus Domino mail server supports any POP3 or IMAP client, including Microsoft Outlook, but it naturally integrates with *IBM Lotus Notes*, the combined desktop client for accessing business e-mail, calendars and applications. Both Lotus Domino and Lotus Notes are well integrated with other IBM products and, through specific add-ins, can provide a wide range of communication and document management functionalities.

Domino mail servers manage also specialized databases for locating users and servers, for message storage and transit, and for collecting statistics that can be accessed by authorized users, like any other Lotus Notes database. Mail databases support full-text indexing, encryption, replication, soft deletions and retention. Administrators can specify properties or policies to limit the use of these features on mail files. Messages in a mail file may be stored in either Notes rich text format or MIME format, depending on user settings.

In addition to a user's primary mail file, users and administrators can replicate mail files to other locations and administrators can create server replicas to provide failover.

Domino mail server also allows the administrator to define and implement simple capture/deletion policies using rules based on some characteristics of the message and on the access log (e.g., arrival time, access times and so on). Lotus supports also client-based capture: in this case, individual users may perform the capture by selecting messages and storing them either in the mail server, a design ated server, or locally in the user’s PC.

To implement a more sophisticated capture/deletion policy, it is necessary to acquire *IBM CommonStore*, a separate product tightly integrated with Lotus Domino and Lotus Notes. CommonStore is actually an “e-mail archiving” product and will therefore be discussed in the next section.

Lotus products are also tightly integrated with *FileNet*, the IBM document and content management product. More precisely, FileNet has a component, named *Email Manager*, which can:

- manage automated e-mail capture;
- monitor e-mail compliance with corporate policies and government regulations;
- launch automatically business processes in response to incoming e-mails;

- manage automated classification of e-mail messages; and
- issue immediate notification that an e-mail has been captured and stored in a centralized repository.

Both Microsoft and IBM products are highly customizable, by means of templates, macros, scripts and proper programs. All basic and popular e-mail functions are available in both products; therefore, the choice between these products is usually more influenced by the existing technical environment than by specific functionalities.

Both of these powerful proprietary solutions enhance standard e-mail management functions with a rich variety of features and provide a very effective integration with other applications. On the one hand, this is a very positive feature, since it improves the quality of the communication and the cooperation of people within the organization. On the other hand, when the communication with the outside world is considered, proprietary components become a negative feature. In an example, some attributes of the message may have to be dropped and a recipient outside the organization may read something that is slightly different from what was sent. For instance, the sender may have highlighted part of the text, and the recipient could miss this information.

Provided proper setup and/or customization, these products may accomplish many of the functionalities needed for implementing e-mail records management policies. In addition, the e-mail application market has recently developed a significant number of specific products for "e-mail archiving" that, by default, provide many of these same records management functionalities

## 7.3 Commercial products for "e-mail archiving"

"E-mail archiving" products were initially developed as "ready to use" solutions for regulatory compliance and legal discovery; hence, their main purpose was to capture all incoming and outgoing corporate e-mail messages and store them in a secure way. However, as this market quickly expanded to a level that is expected to reach one billion dollars in 2008, many functionalities have been added to meet new demands, and current products may satisfy a wide range of requirements spanning from bulk retention to complex records management policies.

It is worth remembering that e-mail protocols require all incoming and outgoing messages to be stored in the e-mail server's transient storage system, where messages are kept until either a user/administrator action or an automated procedure deletes them. Therefore, in principle, initial message capture and storage could be achieved just by implementing a "non deletion" policy at the e-mail server level. However, this approach has several security and performance drawbacks, since e-mail servers are designed to manage transient and short-term storage rather than long-term maintenance or recordkeeping.

"E-mail archiving" products have been designed to fill this gap, that is, to:

- manage a huge number of stored e-mails without affecting e-mail server performances;
- ensure regulatory compliance by capturing messages before they can be modified maliciously or deleted by the recipient;

- allow organizations to implement structured policies for accessing stored e-mail;
- include auditing capabilities to track access to stored records;
- extend active mailboxes by providing user access to the repository via a Web client and through the e-mail client;
- enforce integration between e-mail management systems and r ecords management systems; and
- provide advanced search and knowledge management capabilities.

These functionalities provide a good coverage for the set of requirements that we have outlined above in section 5.2 for the short-term maintenance of e-mail records, since they support bulk e-mail capture and t ransient storage activities, as well as controlled access to stored messages according to security policies. Nevertheless, some products have also functionalities that support also many of the requirements outlined in sections 5.2 and 5.5 for short-term and long-term e-mail maintenance, respectively, either for integration with dedicated records management products (e.g., ERMS) or for performing directly records management tasks.

"E-mail archiving" products are mainly designed for large organizations managing their e-mail through integrated systems. Therefore, practically all vendors support Microsoft Exchange, and a large and increasing number of them support also Lotus Domino.

According to a recent Gartner analysis, Symantec is the market leader, both for ability to execute and for the completeness of vision.[32] Symantec's product, Enterprise Vault, is an integrated content archiving platform supporting e-mail, instant messages, SharePoint and, through third parties add-on modules, popular proprietary repositories (e.g., Bloomberg, BlackBerry).

Besides Symantec, products of several other vendors provide not only for the capture of all e-mail messages, but also allow the definition and the implementation of a retention policy, and give the end-user a view of stored messages similar to an extended mailbox.

Typical functionalities found in these products, interesting for maintenance purposes, are:

- user options for message classification (e.g., Open Text, CommonStore, Message Manager);
- automatic classification capabilities (e.g., CommonStore, EmailXtender, Enterprise Vault, Message Manager);
- cross-user "archive full-text searches" (e.g., Autonomy Zantaz, CommonStore);
- assurance of message authenticity via electronic signature (e.g., HP e-mail archiving, MailMeter);
- disaster recovery features (e.g., NearPoint, Enterprise Vault); and
- storage optimization (e.g., Enterprise Vault).

[32] Carolyn DiCenzo and Kenneth Chin, "Magic Quadrant for E-Mail Active Archiving," Gartner RAS Core Research Note G00157611 (20 May 2008). Accessible via free registration at http://www.mimosasystems.com/html/gartner_mq.htm.

## 7.4 State of the art and trends

The market demand is still driven by the concerns that most organizations and corporations have for regulatory compliance and legal discovery. As the number of exchanged messages shows a steady and robust growing trend, product scalability is a very important issue, and is one of the main reasons why most medium and large organizations use a specific “e-mail archiving product,” in addition to the corporate mail server, to maintain e-mail messages.

However, once the basic requirement of saving e-mail messages, which is vital to many organizations, has been satisfied, the need may arise to extend the e-mail repository to other kinds of content. Many organizations are now interested in implementing a more complete and inclusive records management policy and, thus, are interested in selecting a vendor that will also be able to help with managing records, whether messages, attachments or other kinds of documents.

Classification options are another important issue, but the common approach is still to call for records management capabilities having as input just the e-mail text and RFC 2822 standard fields (from, time, subject, etc.). Consequently, vendors concentrate on efficiently managing stored records and on providing efficient discovery capabilities on unstructured contents. Examples of such features include: advanced search, automatic classification and knowledge management tools.

This market is still aimed essentially at medium and large organizations and, therefore, small organizations and individuals still lack specific products to support their e-mail records management policies and, consequently, may only rely on the functionalities offered by e-mail clients, as we have discussed in section 7.1.

This makes implementing e-mail records management policies a quite difficult task for the individual user and the small organization, since it requires a technical background and professional records management knowledge that they usually lack. Therefore, the only viable solution for these users may be to rely on the more basic e-mail management services offered by e-mail providers or, if possible, on the more qualified expertise of specialized records management companies, a kind of offer that is expected to increase both in volume and in quality in the coming years.

### Acknowledgements

We feel indebted to several colleagues who gave us precious advice while we were preparing and revising this study, primarily the members of InterPARES 3 Team Italy, among whom a special thanks goes to Mariella Guercio and Max Grandi who carefully revised the paper and gave us several valuable suggestions. We would also like to thank Randy Preston for his numerous remarks and suggestions, some of which we have appreciated and taken into account in the last version of the paper.

## Appendix A - Requirements for systems to manage and preserve e-mail records

This appendix discusses some of the most important references that should be taken into account in designing transient storage, corporate recordkeeping and/or permanent preservation systems that are capable of managing or preserving e-mail records. Most of these requirements have been indeed taken into account in this report when discussing the corresponding issues.

However, to improve reading, we decided to avoid in sections 5 and 6 notes and detailed references. In this appendix, we systematically analyze the references, discuss their purposes and relevance, and specifically refer to the requirements they contain that are relevant for e-mail management.

### DoD 5015-02- STD - Electronic Records Management Software Applications Design Criteria Standard (2007)

The DoD 5015.2, "*Department of Defense Records Management Program*," is a vendor standard for the US Department of Defense, issued on April 1997, which provides implementation and procedural guidance for the management of records in the Department of Defense. A second version was developed in 2002, and the third version was published in April 2007.

The standard sets forth mandatory baseline functional requirements for Records Management Application (RMA) software used by the DoD in implementing their records management programs. More precisely, it defines the required system interfaces and search criteria that RMAs shall support; and describes the minimum records management requirements that must be met based on current US National Archives and Records Administration (NARA) regulations.

The DoD 5015-02-STD provides complete and thorough requirements and has a relevant influence on the development of commercial products. Moreover, DoD has established a compliance testing process managed by the Joint Interoperability Test Command (JITC) of the Defense Information Systems Agency (DISA). Therefore, since all systems acquired by the DoD and by the US Federal Government have to undergo the compliance testing procedure, the requirements in the standard were designed to be reasonably met by commercial product.

The PDF version of the 2007 standard is currently available from:

http://jitc.fhu.disa.mil/recmgt/p50152stdapr07.pdf (*visited 9-2008*)

The third version of DoD 5015-02-STD contains several requirements that specifically pertain to the management of e-mail messages as records, and other general requirements that concern other issues we discuss in this report, as security, access control and audit.

- *Message capture and classification*

  Section C2.2.4 gives a set of mandatory requirements specific to e-mail records, relating to the capture process, the user interface and the metadata extraction. According to these requirements, the message can either filed as a whole or attachments can be stored as separate records and linked to the main record. The section contains also a table (Table C2.T4 Transmission and Receipt Data) where the data that must be captured and their corresponding metadata are listed.

- *Search and retrieval*

  Search and retrieval requirements are discussed in section C2.2.7.8, which, besides general requirements on query capabilities, specifically calls for the "capability for filed e-mail records to be retrieved back into a compatible e-mail application for viewing, forwarding, replying, and any other action within the capability of the e-mail application" (C2.2.7.8.7).

- *Access control and audit*

  These issues are discussed in sections C2.2.8 and C2.2.9. More specifically, a very detailed access control scheme is given (Table C2.T6 – *Mandatory Authorized Individual Requirements*), where access rights are defined for the roles of *Applications administrator*, *Records manager* and *Privileged user*. Though mostly not specific, the scheme should be considered in the design and configuration of "e-mail archiving applications."

- *Metadata*

  Metadata are discussed in section C5.1, and specifically e-mail metadata in section C5.1.6.3, where mandatory metadata are specified in Table C5.2 – *Record Level E-mail,* together with the indication of their source.

- *Additional requirements*

  Further requirements relate to the management of e-mail messages from wide-area networks other that the Internet, that should be treated in the same way (C2.2.12.2), and to the management of e-mail distribution lists.

### MoReq2 – Model Requirements for the Management of Electronic Records (2008)

This specification is a new, and more detailed, version of an original specification (MoReq) that was issued and published in 2001 under funding of the European Commission. MoReq had been widely used in Europe and beyond by prospective users of electronic records management as a model specification in procuring Electronic Records Management Systems, and by software suppliers as a guide to the development process.

MoReq2 was prepared for the European Commission by Serco Consulting, a U K consulting firm, with financing from the European Union's IDABC programme. The development process was overseen by the European Commission and by the DLM (Document Lifecycle Management) Forum.

The goal of MoReq2 is rather ambitious, since, besides providing extended functional requirements, it aims also at "ensuring that the functional requirements are testable and developing test materials to enable products to be tested for compliance with the requirements."

The PDF version of MoReq2 is currently available from:

> http://www.cornwell.co.uk/moreq2/MoReq2_body_v1_04.pdf (*visited 9-2008*)

MoReq2 contains both requirements that specifically pertain to the management of e-mail messages as records, and other general requirements that concern other issues we discussed in this report, such as security, access control and audit.

- *Message capture and classification*

  The capture process is discussed in great detail. General requirements, most of which are relevant for the e-mail case, are discussed in section 6.1, and specific requirements for e-mail in section 6.1.3. Both the system-based and the user-based schemes are proposed, and several important issues are defined as the handling of attachments and the problem of linking messages of the same thread.

- *Search and retrieval*

  These issues are discussed in section 8. Only general requirements are given, but the discussion is rather detailed, especially on search criteria where the use of thesauri and ontologies is thoroughly discussed. Presentation issues are discussed in section 8.2.

- *Access control and audit*

  These issues are discussed in section 4. A general discussion is provided, but no issues specific to e-mail management are presented. Nevertheless, it is a valuable reference since it thoroughly discusses the matter, including such issues as the ownership of records and the management of groups and roles.

- *Metadata*

  Metadata are discussed in much detail in Appendix 9, where a rich set of e-mail specific metadata is defined. For each metadata, the source, the population criteria and the use conditions are specified. This set of metadata has been taken into account in writing section 5.4 of this report, and is actually a subset of the metadata in Table 2 of this report.

- *Additional requirements*

  Further discussion refers to the integration of fax servers with e-mail servers (section 10.2) to file faxes which are sent as e-mail attachments, and to the integration between ERMS and e-mail system to allow sending records from within the ERMS (section 11.1).

### UK Public Record Office - Functional Requirements for Electronic Records Management Systems (2002)

This specification is composed of four chapters that are published as separate documents. At least two of them are relevant to e-mail management:

- Part 1 – *Functional requirements*, currently available from:

  http://www.nationalarchives.gov.uk/documents/requirementsfinal.pdf (*visited 9-2008*)

- Part 2 – Metadata standard, currently available from:

  http://www.nationalarchives.gov.uk/documents/metadatafinal.pdf (*visited 9-2008*)

More specifically, the *Functional requirements* discuss the capture of e-mail messages (section A.2), including declaration metadata extraction, export and transfer of e-mail records (section A.4), and other more general issues about access control, audit and usability. The M etadata standard proposes a general set of metadata, and carefully discusses the problem of extracting metadata for e-mail messages.

### DCC-Digital Curation Manual

The Digital Curation Manual is published by the DCC (*Digital Curation Centre*), a project funded by JISC (*Joint Information Systems Committee*), which supports United Kingdom post-16 and higher education and research.

The Digital Curation Centre has the aim of supporting UK institutions which store, manage and preserve records and the documentary heritage created in digital form, to help ensure their enhancement and their continuing long-term use.

Currently, eleven chapters of the DCC Digital Curation manual are available, and can be downloaded from:

http://www.dcc.ac.uk/resource/curation-manual/chapters/ (*visited 9-2008*)

The chapter *Curating E-Mails: A life-cycle approach to the management and preservation of e-mail messages,* by Maureen Pennock, published in 2006, contains a very interesting survey of the problems connected with e-mail management.