EMAIL

Ricardo Azevedo
ricardo.azevedo@ua.pt

Cofinanciado por:

Sumário

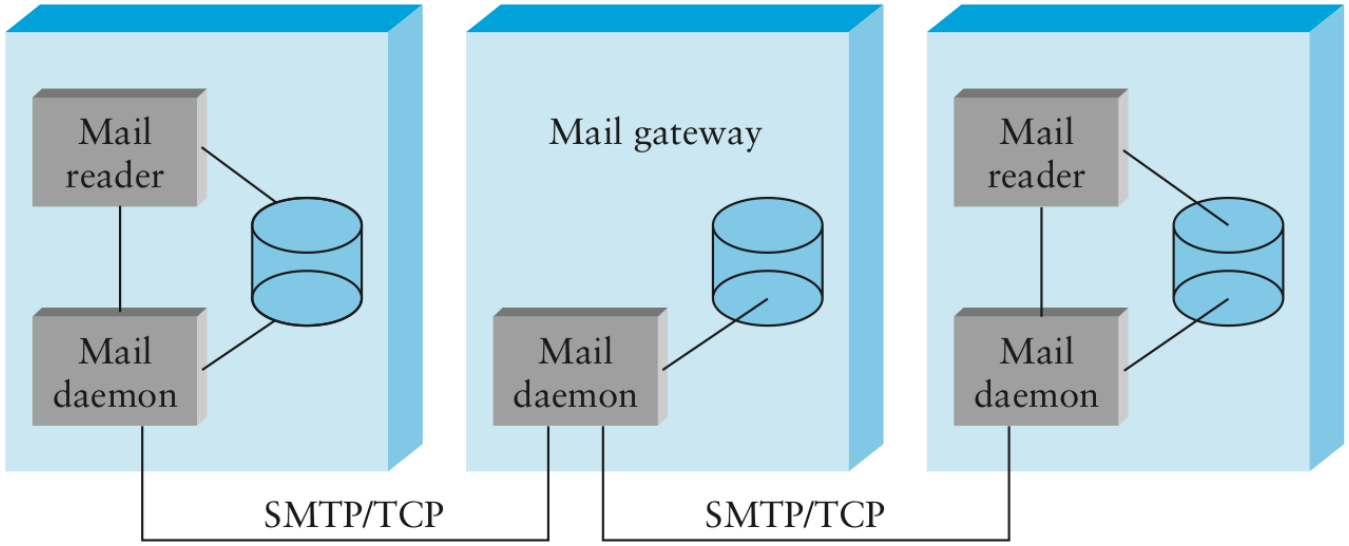
- Funcionamento dos serviços de email
- Protocolos de email
- Formato das mensagens de email
- Portos usados
- Questões de segurança

Serviço de email

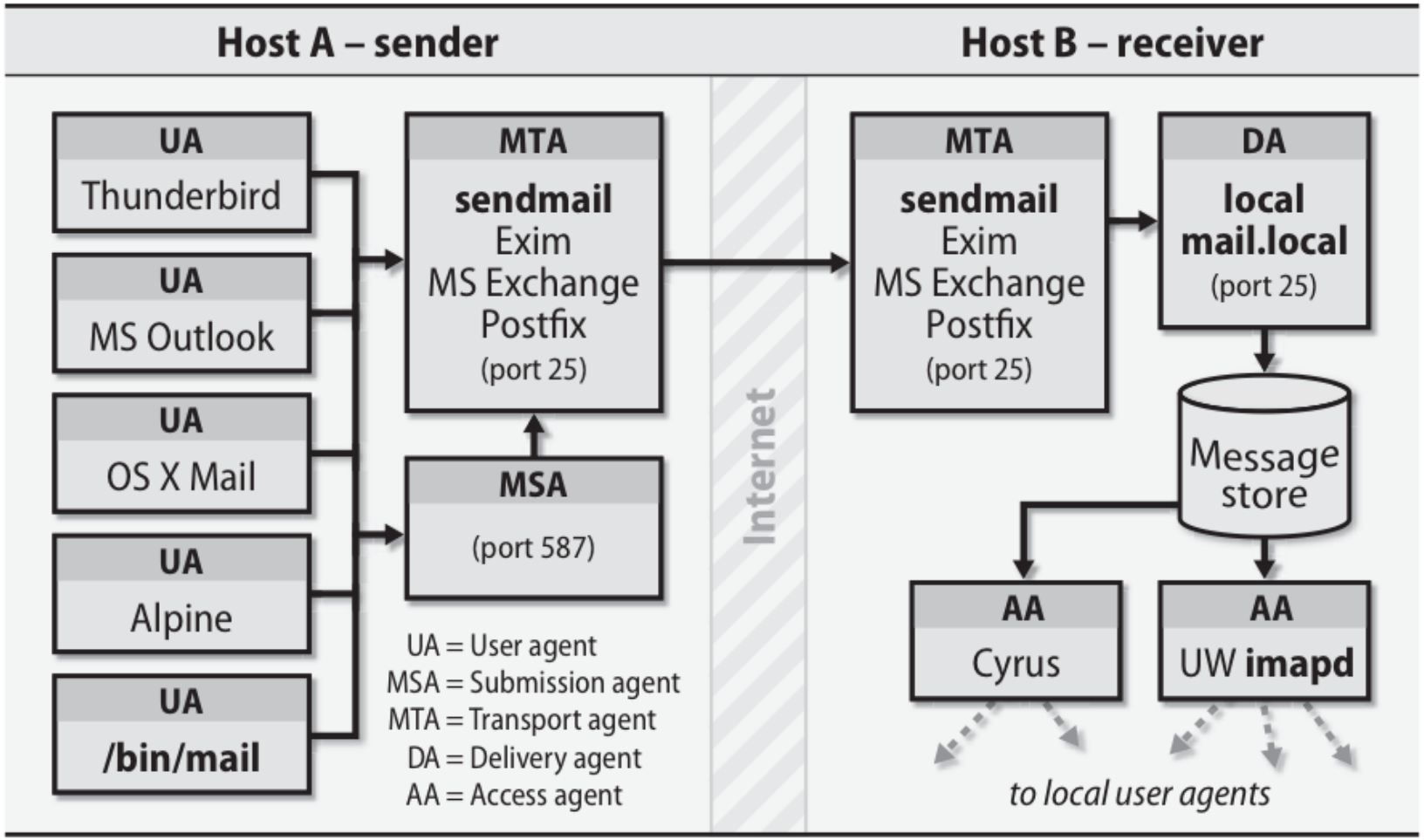
- Começou por ser mecanismo de comunicação entre utilizadores da mesma mainframe.
- Foi estendido para comunicação entre utilizadores de mainframes com mesmo SO.
- Interoperabilidade proposta em 1973 (RFC 561)
- SMTP proposto em 1982 (RFC 821)

Funcionamento

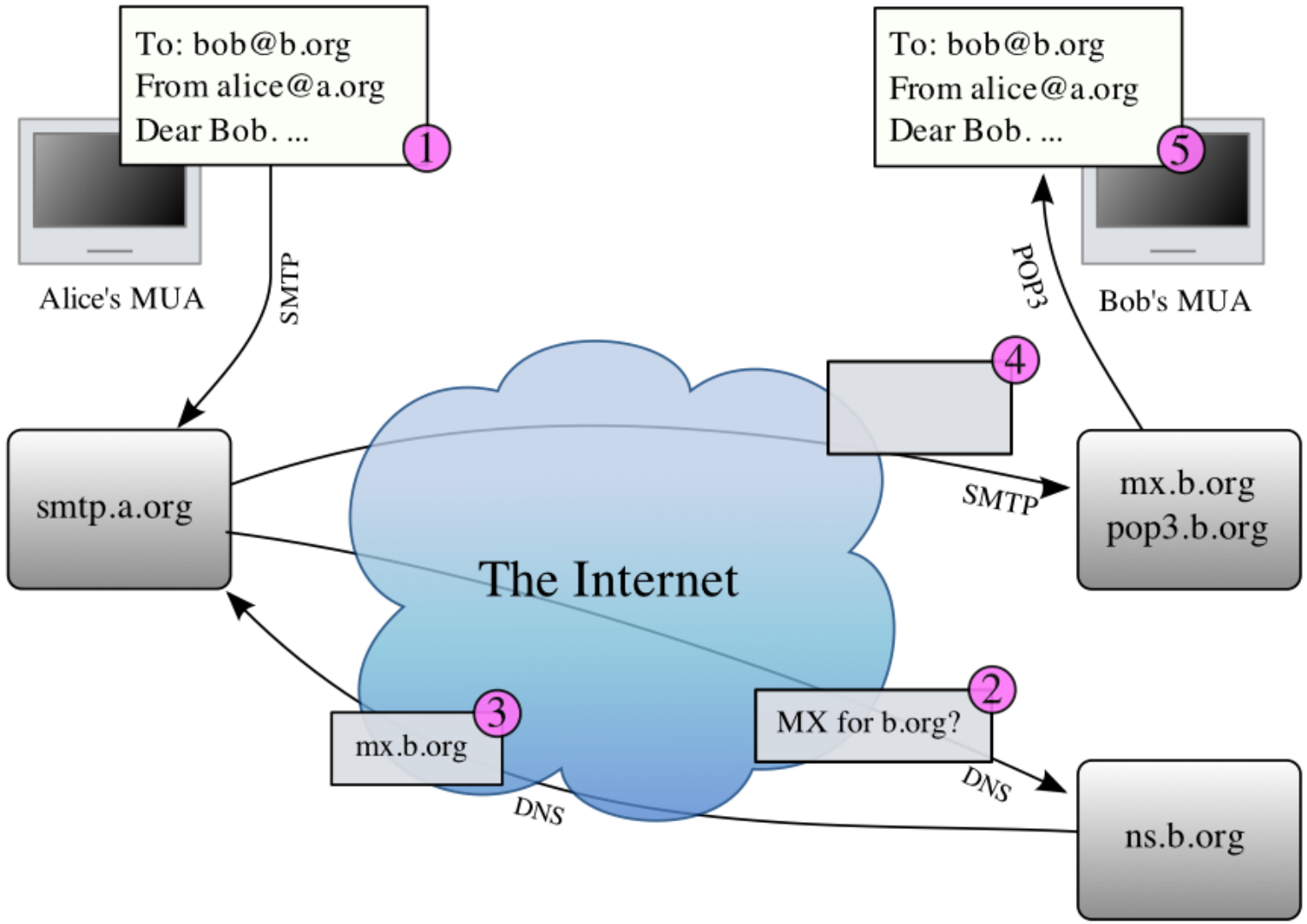
- Utilizadores servem-se de aplicações que comunicam com os seus Mail User Agent (MUA).
- Serviço efectua transporte de correio até destino.



Detalhes de funcionamento



Sequencia de operações

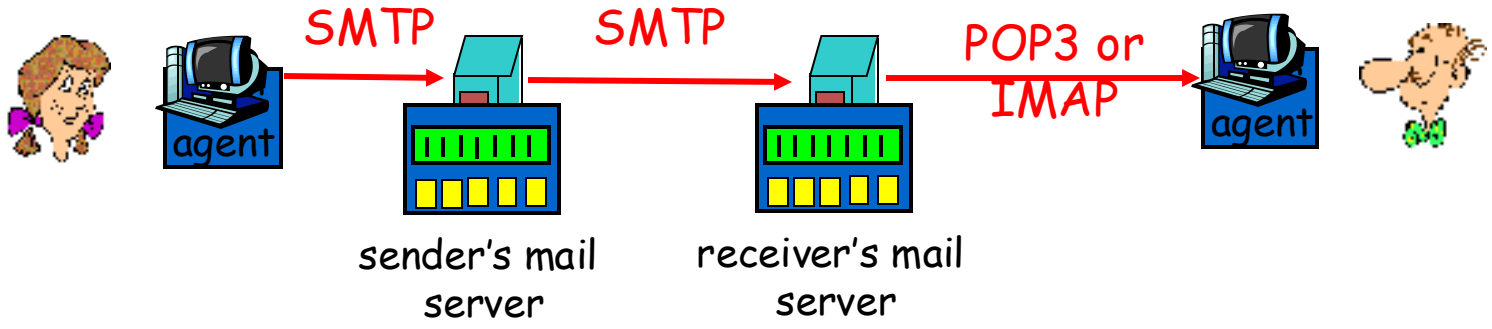


Protocolos de transporte de correio

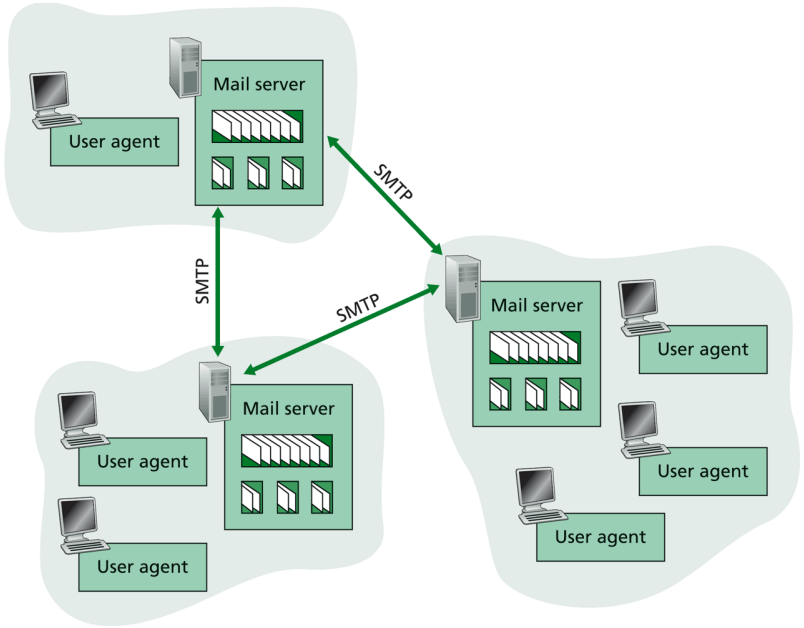
- Envio:
  - Simple Mail Transport Protocol:
    - Inicialmente proposto na RFC 821 actualmente na RFC 5321
- Recepção:
  - Post Office Protocol (POP)
  - IMAP (Internet Mail Access Protocol)

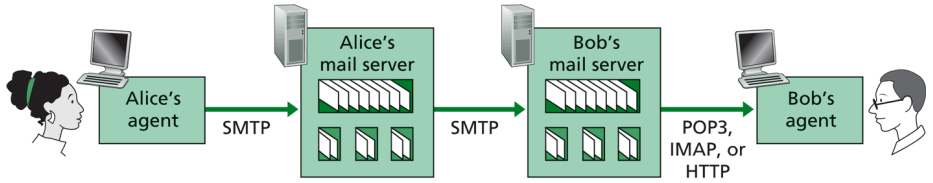
Protocolos de email

- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Hotmail , Yahoo! Mail, etc.



SMTP: envio de correio electrónico






```
S: 220 mr1.its.yale.edu
C: HELO cyndra.yale.edu
S: 250  Hello cyndra.cs.yale.edu, pleased to meet you
C: MAIL FROM: <spoof@cs.yale.edu>
S: 250 spoof@cs.yale.edu... Sender ok
C: RCPT TO: <yry@yale.edu>
S: 250 yry@yale.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Date: Wed, 23 Jan 2008 11:20:27 -0500 (EST)
C: From: "Y. R. Yang" <yry@cs.yale.edu>
C: To: "Y. R. Yang" <yry@cs.yale.edu>
C: Subject: This is subject
C:
C: This is the message body!
C: Please don't spoof!
C:
C: .
S: 250 Message accepted for delivery
C: QUIT
   221 mr1.its.yale.edu closing connection
```

POP and IMAP para recepção de correio

- These are protocols for how to deal with a mailbox server
- To SEND mail, both POP and IMAP clients use SMTP

- POP and IMAP clients need configuration:
  - mailbox server
  - SMTP server

Client/Server – 1 de 3 modelos

- Offline (POP3)
  - Cliente liga-se ao servidor e puxa todo o email
  - Tudo fica alojado no cliente

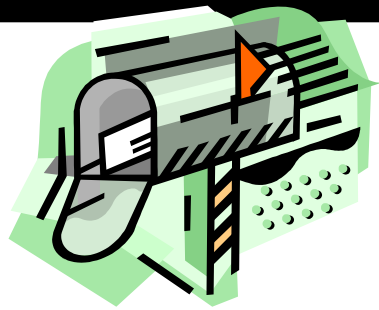
- Online (IMAP original)
  - Client liga-se ao servidor em cada transacção
  - Tudo fica no servidor

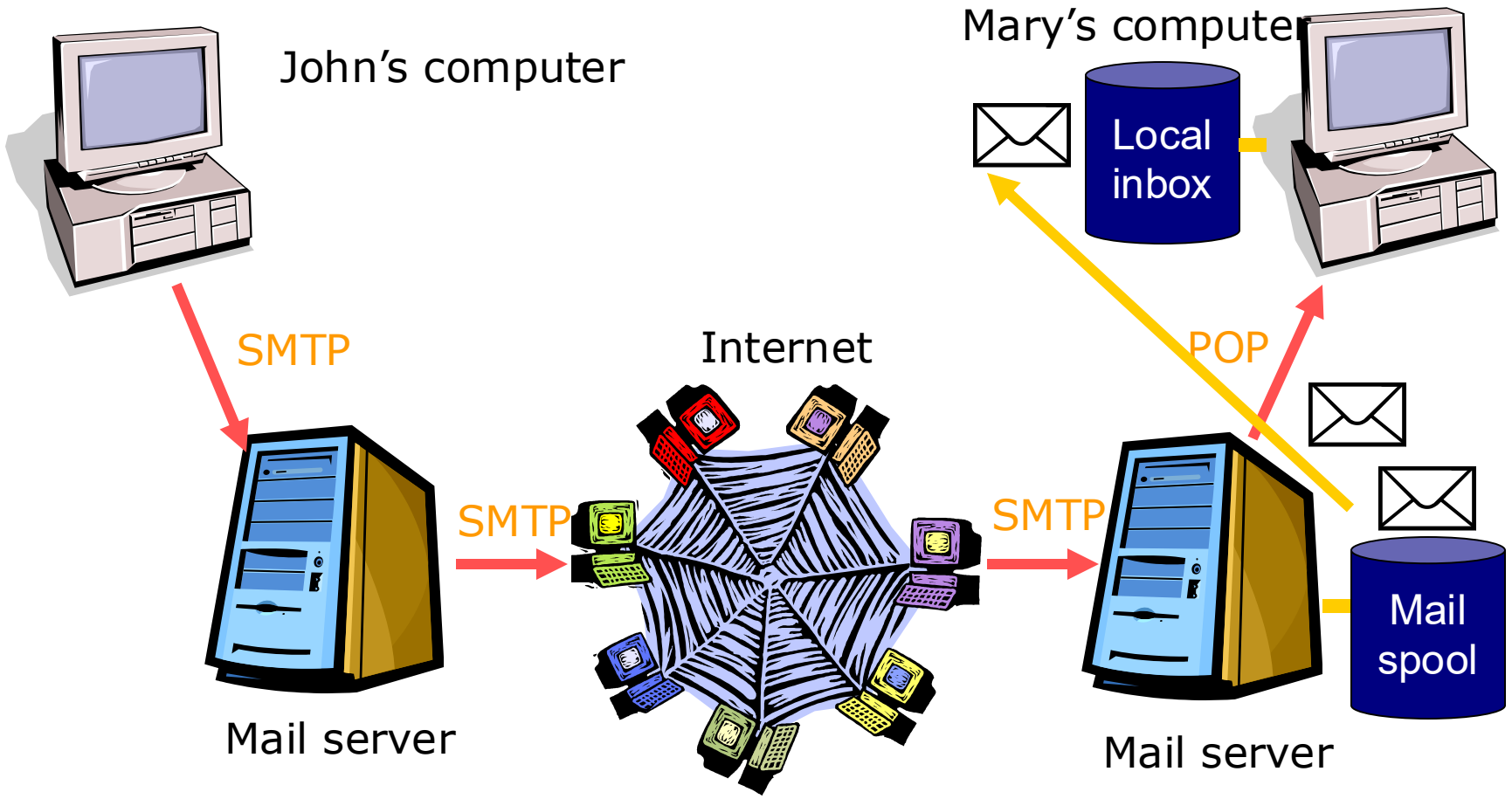
- Desligado (IMAP)
  - Armazenamento feito no cliente e no servidor
  - Server é sempre prevalente e cliente tem que se sincronizar com ele.

POP - Post Office Protocol

- POP client liga-se ao servidor e copia tudo para repositório local.
- Suporta leitura de correio offline
- Interacção típica com o servidor:
  - Liga-se ao servidor
  - Recebe todas as mensagens
  - Armazena mensagens em repositório local
  - Apaga mensagens do servidor
  - Desliga-se do servidor
- Pode ser configurado para manter mensagens no servidor.

Ilustração acerca de POP





Sessão POP

```
$ telnet/port=110 mail.opus1.com
Trying... Connected to MAIL.OPUS1.COM.

+OK cello.Opus1.COM MultiNet POP3 Server Process V4.0(1) at Fri 20-
Sep-96 3:21PM-MST
user trumbo
+OK User name (trumbo) ok. Password, please.
pass thisismypasswordincleartext
+OK 3 messages in folder NEWMAIL (V4.0)
list 2
+OK 2 7124
stat
+OK 3 14749
last
+OK 0
quit
+OK POP3 MultiNet cello.Opus1.COM Server exiting (3 NEWMAIL messages
left)

Connection closed by Foreign Host
$
```

‘list’ gives individual message size in bytes

‘stat’ gives total message size in bytes

Protocolo POP3: acesso ao email

Authorization phase

- client commands:
  - **`user:`** declare username
  - **`pass:`** password
- server responses
  - **`+OK`**
  - **`-ERR`**

Transaction phase, client:

- **`list:`** list message numbers
- **`retr:`** retrieve message by number
- **`dele:`** delete
- **`quit`**

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

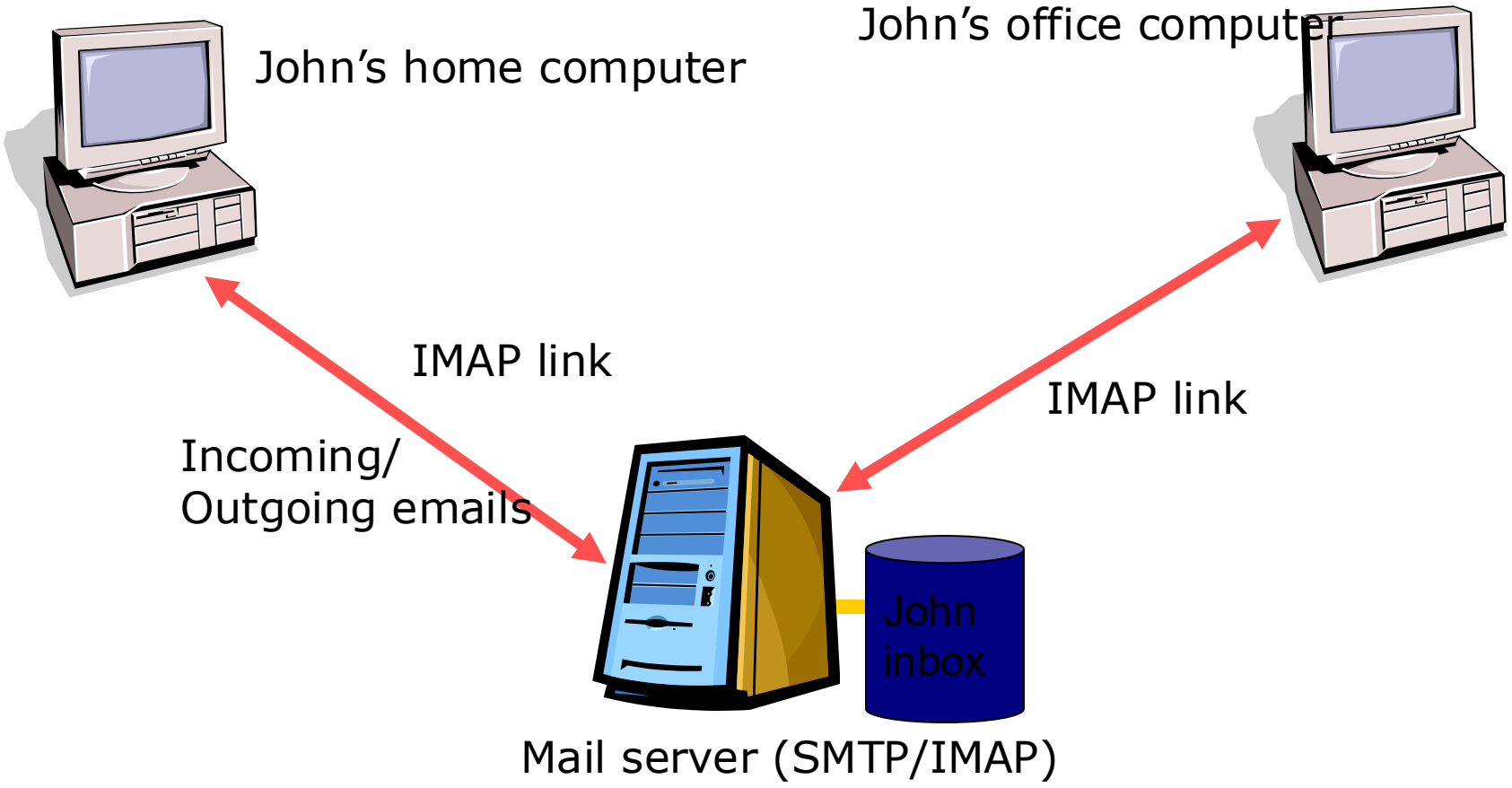
```
C: list
 S: 1 498
 S: 2 912
 S: .
 C: retr 1
 S: <message 1 contents>
 S: .
 C: dele 1
 C: retr 2
 S: <message 1 contents>
 S: .
 C: dele 2
 C: quit
 S: +OK POP3 server signing off
```

```
%telnet <netid>.mail.yale.edu 110
%openssl s_client -connect pop.gmail.com:995
```

Interactive Mail Access Protocol IMAP

- Aceita os modos:On-line, off-line, or disconnected mode operation
- Permite o controlo de pastas de qualquer local
- Permite multiplas caixas num mesmo servidor
- Permite a criação e alteração de pastas no servidor
- Permite procuras em cima do servidor
- Permite acesso ao servidor a múltiplos clientes

Leitura de correio IMAP



IMAP 4

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

OK Dovecot ready.

1 login john@example.com summersun

1 OK Logged in.

list "" "*”

* LIST (\HasNoChildren) "." "INBOX"

2 OK List completed. 3 select "INBOX” * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)

* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft *)] Flags permitted.

* 1 EXISTS

* 0 RECENT

* OK [UIDVALIDITY 1180039205] UIDs valid

* OK [UIDNEXT 3] Predicted next UID

3 OK [READ-WRITE] Select completed.

4 fetch 1 all

* 1 FETCH (FLAGS (\Seen) INTERNALDATE .....

4 OK Fetch completed.

5 fetch 1 body[]

* 1 FETCH (BODY[] {474}

Return-Path: <steve@example.com>

X-Original-To: john@example.com

Delivered-To: john@example.com

Received: from example.com (localhost [127.0.0.1])

by ... (Postfix) with ESMTP id 692DF379C7

for <john@example.com>; Fri, 18 May 2007 22:59:31 +0200 (CEST)

Message-Id: <...>

Date: Fri, 18 May 2007 22:59:31 +0200 (CEST)

From: steve@example.com

To: undisclosed-recipients:;

Hi John,

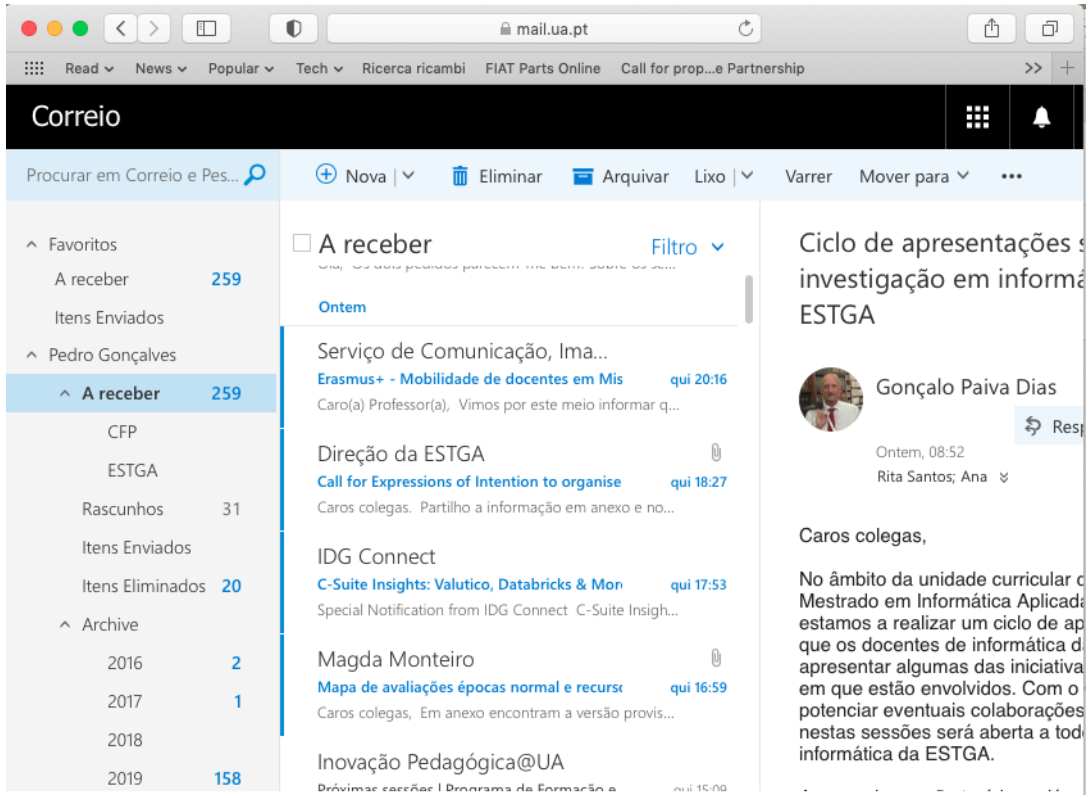
just wanted to drop you a note.

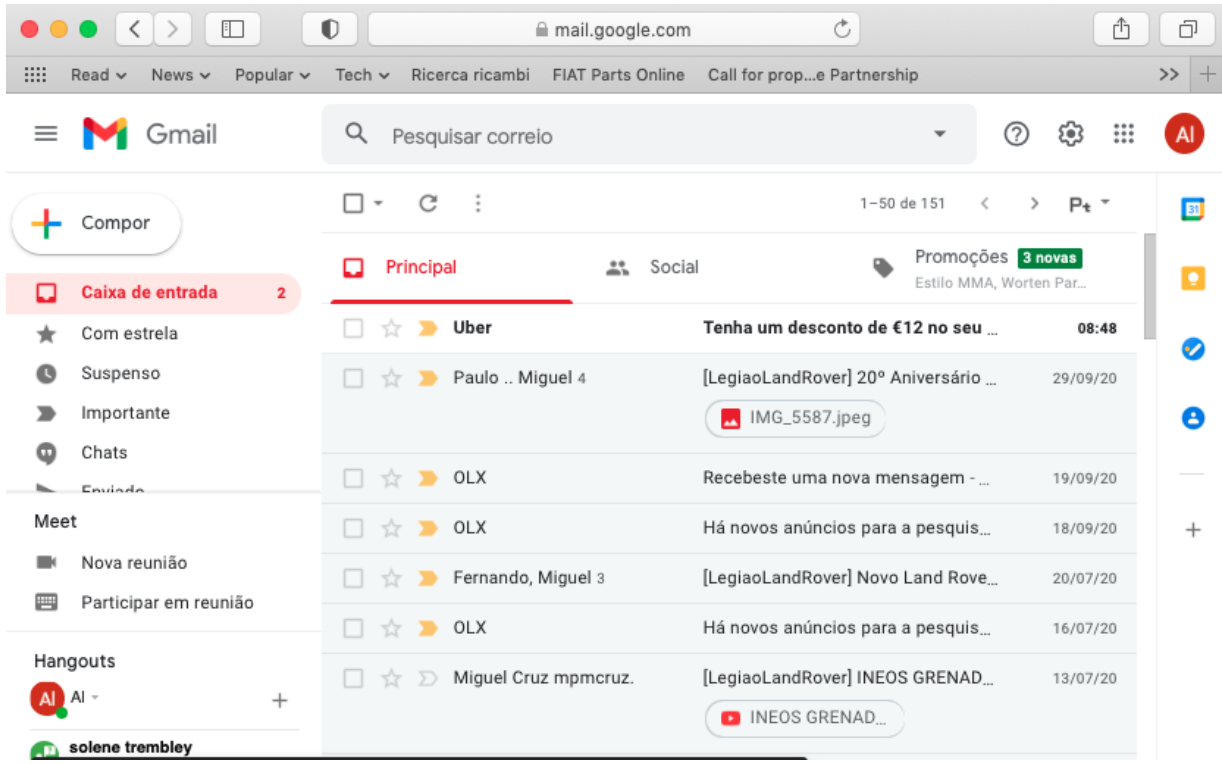
)

5 OK Fetch completed.

Web mail

- Acesso ao correio electrónico através do browser
- Servidor web integrado com o servidor de SMTP

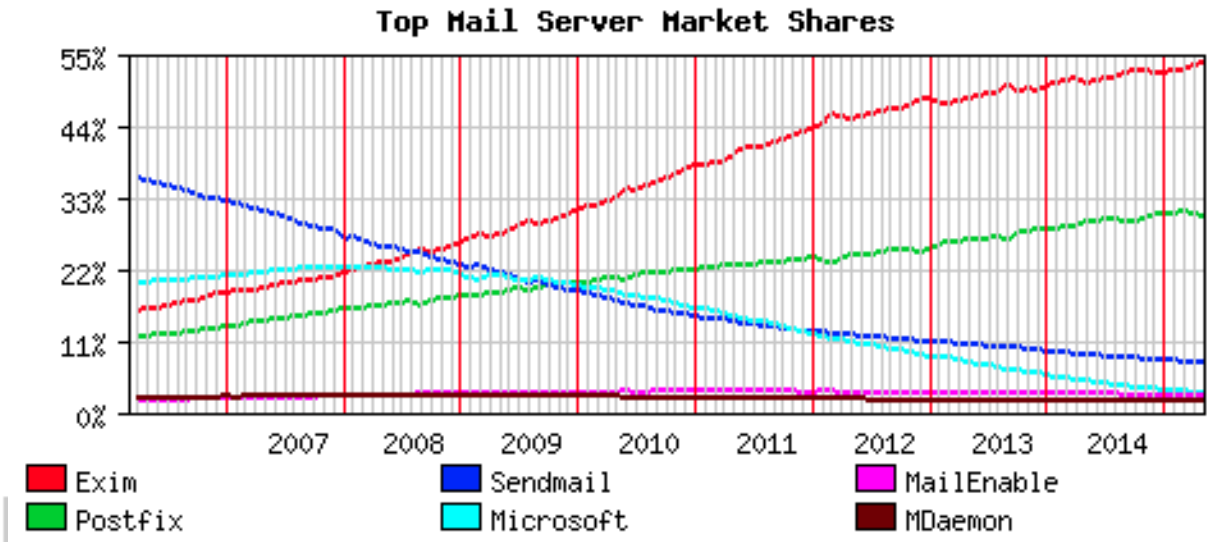




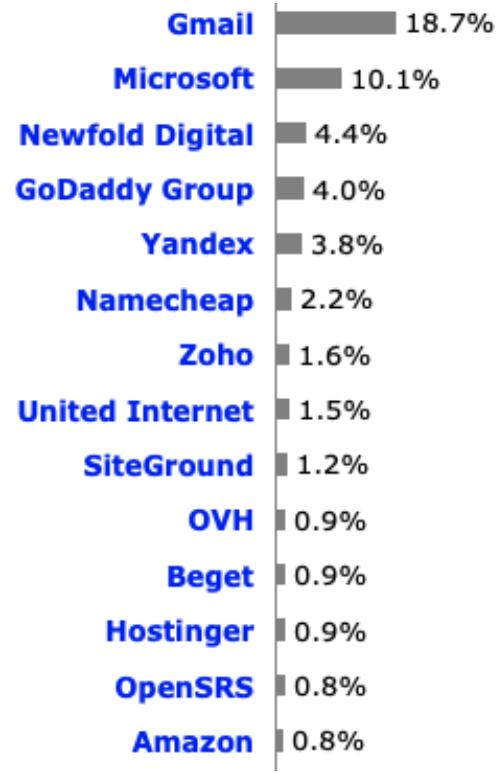
Portos

- SMTP
  - 25
  - Sec SMTP: 465
- POP:
  - 110
  - sPOP: 995
- IMAP:
  - 143
  - secIMAP: 993
- Webmail: 80

Share de utilização de email servers



Exim	581,997	53.53%
Postfix	328,766	30.24%
Sendmail	82,400	7.58%
Microsoft	30,787	2.83%
MailEnable	28,862	2.65%
MDaemon	17,277	1.59%
IMail	4,014	0.37%
CommuniGate Pro	2,461	0.23%
Lotus Domino	2,261	0.21%
WinWebMail	1,664	0.15%



http://www.securityspace.com/s_survey/data/man.201504/mxsurvey.html

https://w3techs.com/technologies/overview/email_server

Soluções integradas de email

- Citadel (citadel.org)
- Zimbra (zimbra.com)
- Kerio MailServer (kerio.com)
- Communigate Pro (communigate.com)
- MS Exchange
- OpenExchange

Formato das mensagens

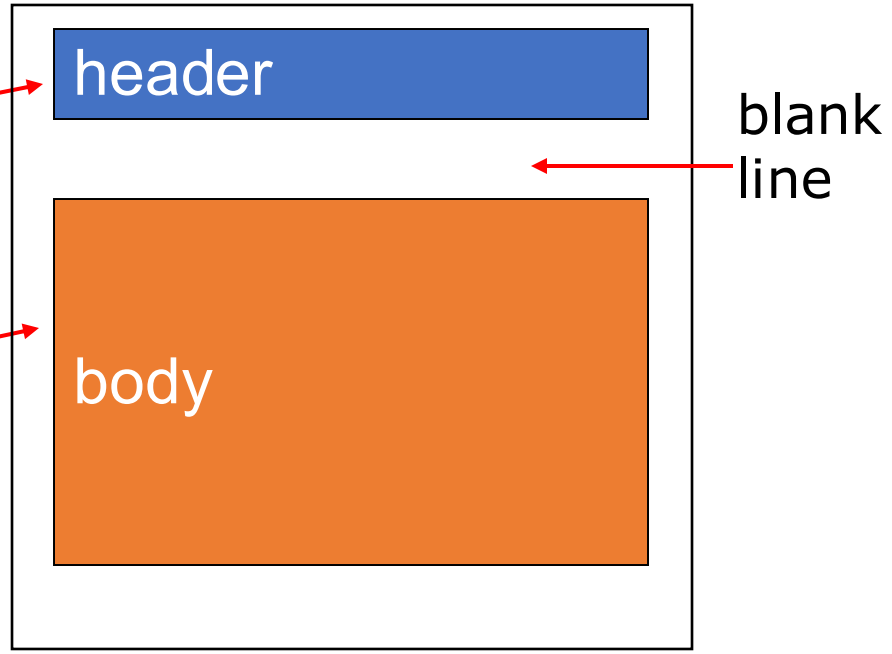
Anatomia de mensagem de email

- Contém:
  - Envelope (nem sempre visível)
  - Cabeçalho: Definido na RFC 5322
    - Campos : From, To, Subject, Date, Message-ID
  - Corpo: Definido nas RFC 2045 a 2049
    - Inicialmente em texto (ASCII 7 bits)
    - Pode incluir corpo escrito em HTML
    - Inclui um conjunto de elemtos segundo uma norma designada de MIME
    - Formato HTML é muitas vezes usado como técnica de phishing

Formato das mensagens de Mail

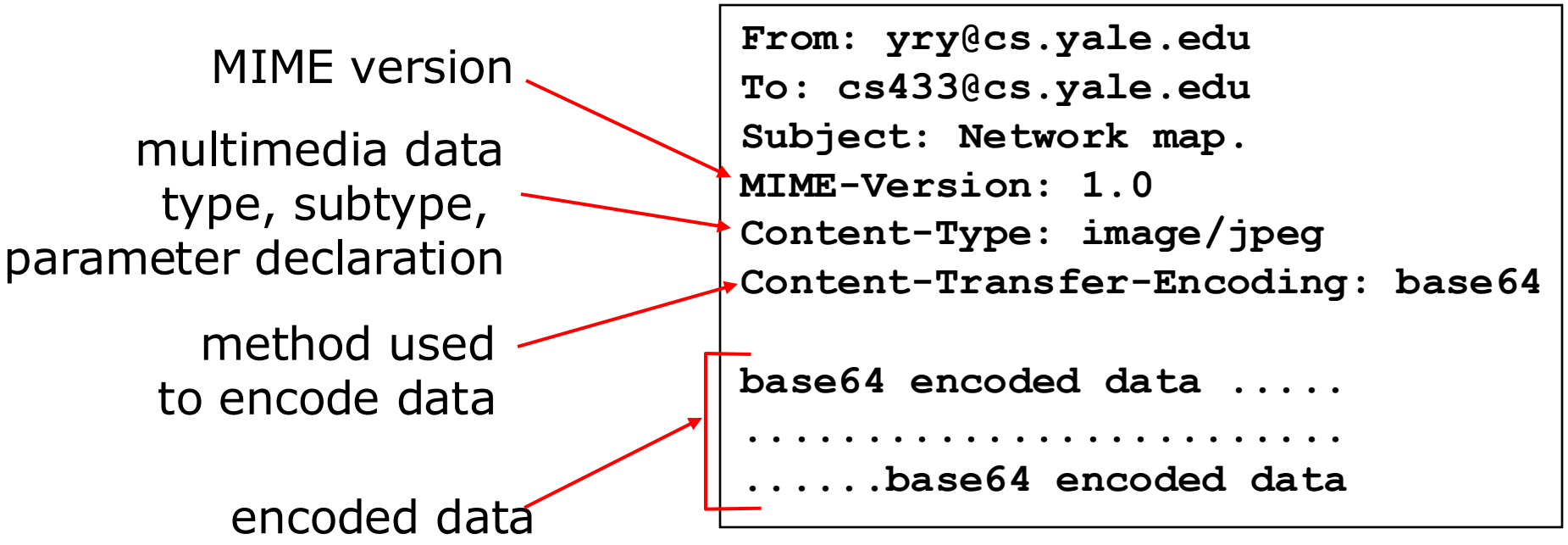
SMTP: protocolo para troca de mensagens de email
RFC 822: standard para formato da message:

- Header,
  - To:
  - From:
  - Subject:
- Body
  - A mensagem em caracteres ASCII



Formato da Mensagem : Multimedia Extensions

- MIME: extensão multimedia para email, RFC 2045, 2056
- Linhas adicionais no header declaram o MIME content type



Multipart Type: como funciona o Attachment

```
From: yry@cs.yale.edu
To: cs433@cs.yale.edu
Subject: Network map.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789

--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain

Hi,
Attached is network topology map.
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
--98766789--
```

Perguntas

- Qual a intervenção do serviço de nomes no funcionamento dos serviços de correio?

- Qual a função do registo do tipo MX do DNS? Qual a consequência de um erro no valor do registo MX?

- Proponha um mecanismo simples para detectar forged emails.

- Em que consistem as blacklists dos serviços de emails? Que pode fazer para não ser incluido nessas listas?

Referências

- http://www.spamhaus.org/
- http://workaround.org/ispmail/lenny

Phishing & Spam & …

The threat landscape has evolved drastically in 2024-2025.

> 193,000+ phishing complaints filed in 2024 alone.

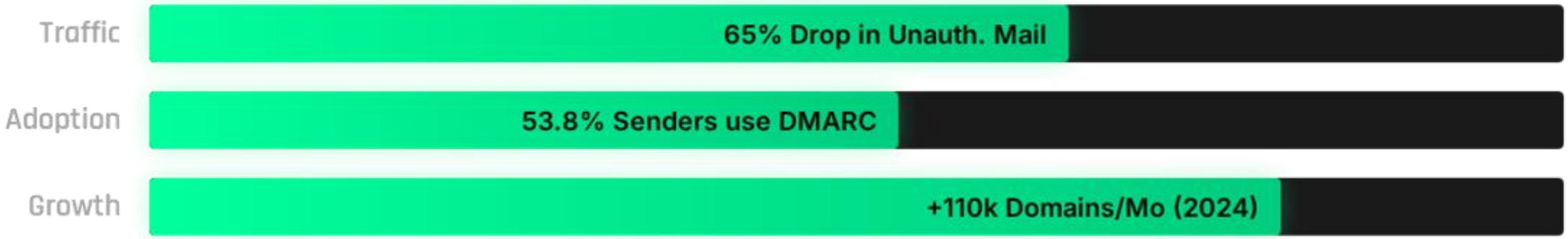
> $1.14 Million average cost per successful phishing incident.

> AI-Driven: 40% of Business Email Compromise (BEC) attacks now utilize Generative AI for hyper-realistic targeting.

> Volume: Phishing losses have quadrupled compared to previous years.

Industry response and adoption

Mandates from Google and Yahoo have driven a massive shift in protocol adoption.



DMARC adoption rates surged by 11% in a single year due to new requirements.

Sources: Google Security Blog, Valimail 2024

The protocols

SPF

Sender Policy Framework

Defines "Who is allowed to send".

DKIM

Domain Keys Identified Mail

Verifies "Content integrity".

DMARC

Domain-based Authentication

Enforces "What to do with failures".

SPF (Sender Policy Framework)

Think of SPF as a security guard with a guest list.

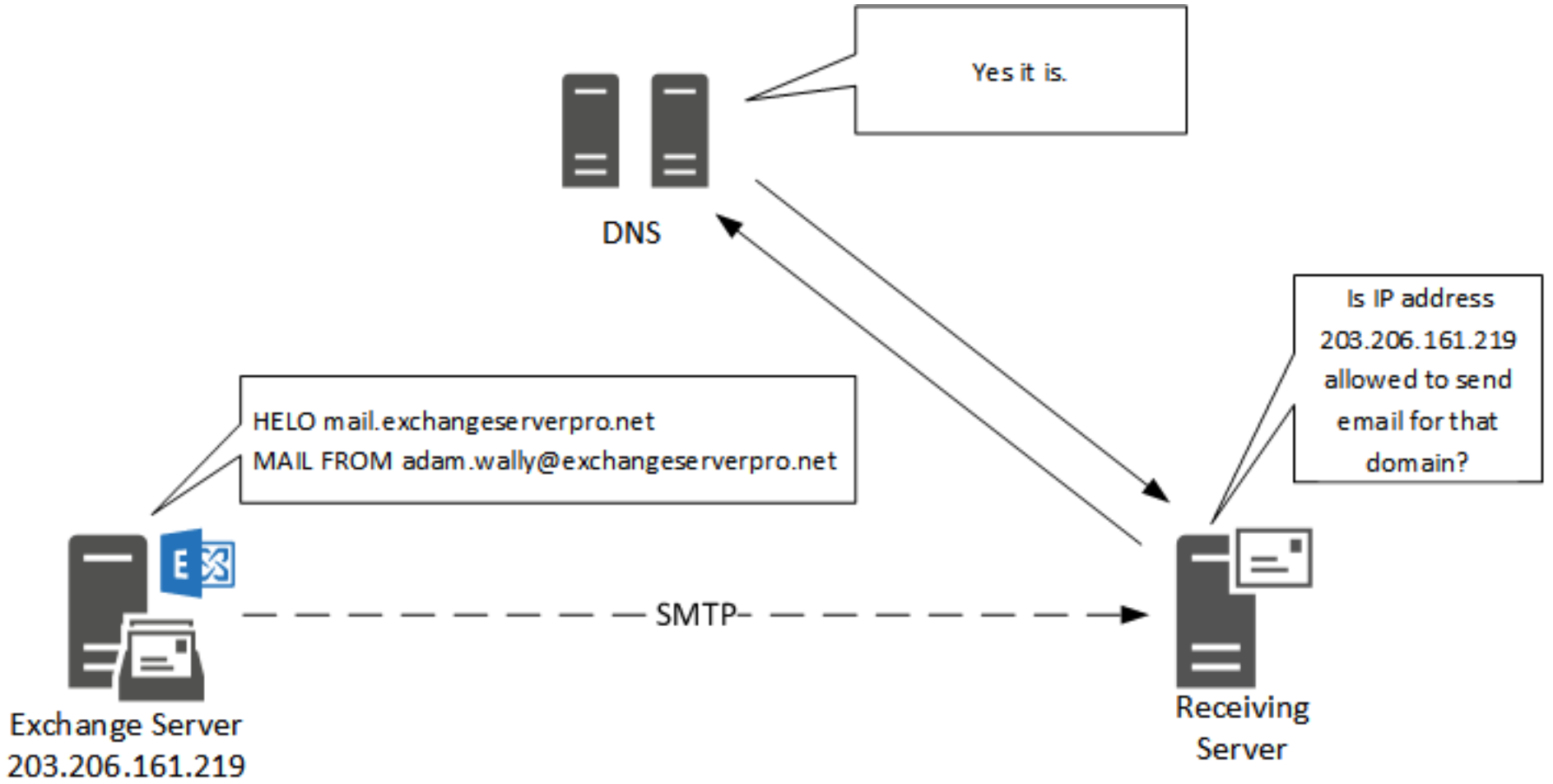
> The "Guest List" is a DNS text record.

> It lists every IP address authorized to send mail for your domain.

> If a server tries to send mail and isn't on the list, the guard (receiving server) flags it.

> Crucial: It only checks the "Envelope Sender", not the visible "From" address.

SPF Flow



src: https://practical365.com/a-sender-policy-framework-spf-primer-for-exchange-administrators/

SPF - Configuration example

THE DNS RECORD

```
v=spf1 include:_spf.google.com
ip4:192.168.1.50 ~all
```

This single line of text authorizes Google Workspace and one specific office IP to send mail.

BREAKDOWN

- v=spf1: Identifies the record type.
- include:: Trusts a 3rd party (e.g., Google, Mailchimp).
- ip4:: Trusts a specific static IP.
- ~all: Soft Fail. (Use -all for Hard Fail/Reject).

DKIM (Domain Keys Identified Mail)

TAMPER-PROOFING

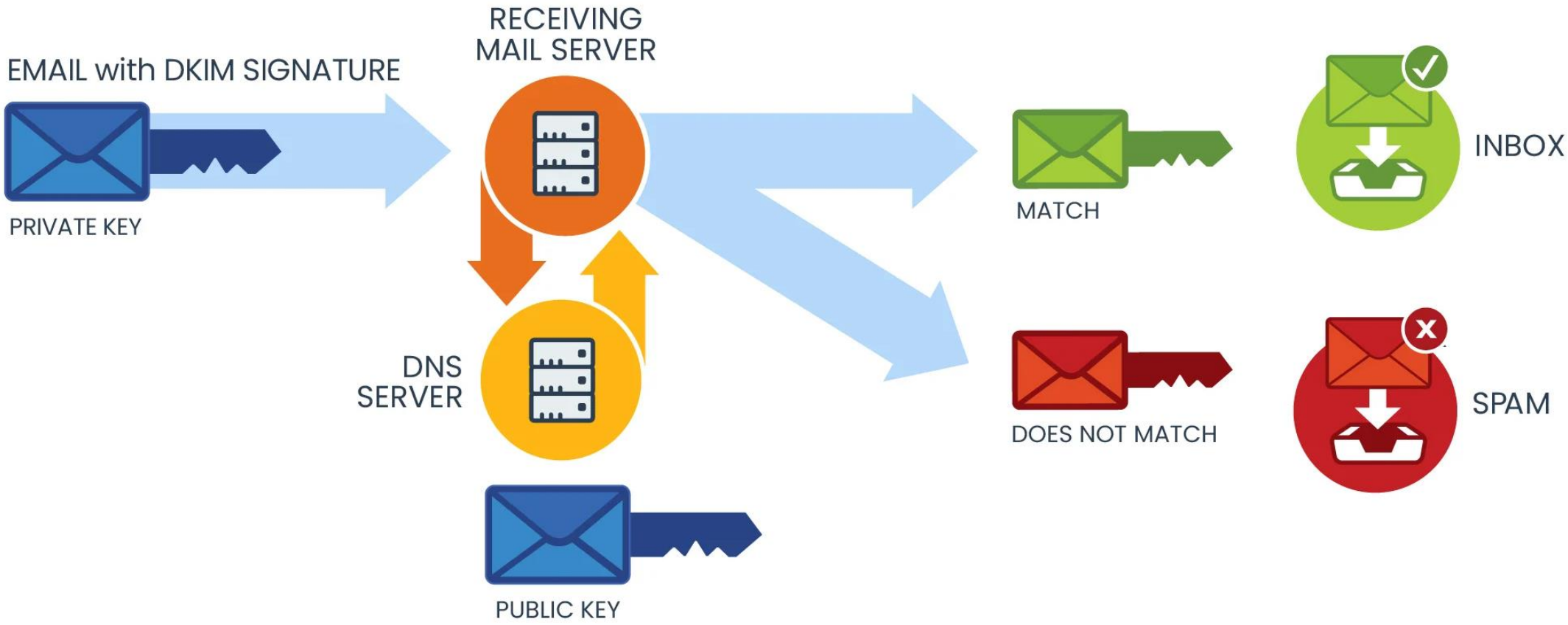
DKIM adds a cryptographic signature to the email header.

> Ensures Integrity: Proves the email hasn't been altered in transit.

> Uses Public/Private Key cryptography.

> Like a wax seal: if the seal is broken (hash doesn't match), the recipient knows the message was tampered with.

DKIM Flow



src: https://www.emailonacid.com/blog/article/email-deliverability/what-is-dkim-everything-you-need-to-know-about-digital-signatures/

DKIM - Configuration example

1. DNS RECORD (PUBLIC KEY)

```
google._domainkey TXT "v=DKIM1; k=rsa;
p=MIIBIjANBgkq..."
```

Selector: "google" (identifies which key to use).

Key: The long string "p=..." is the public key.

2. EMAIL HEADER (SIGNATURE)

```
DKIM-Signature: v=1; a=rsa-sha256; d=example.com;
s=google; bh=...
```

d=: Signing domain.

s=: Selector used to find the key in DNS.

DMARC (Domain-based Authentication)

THE POLICY ENGINE

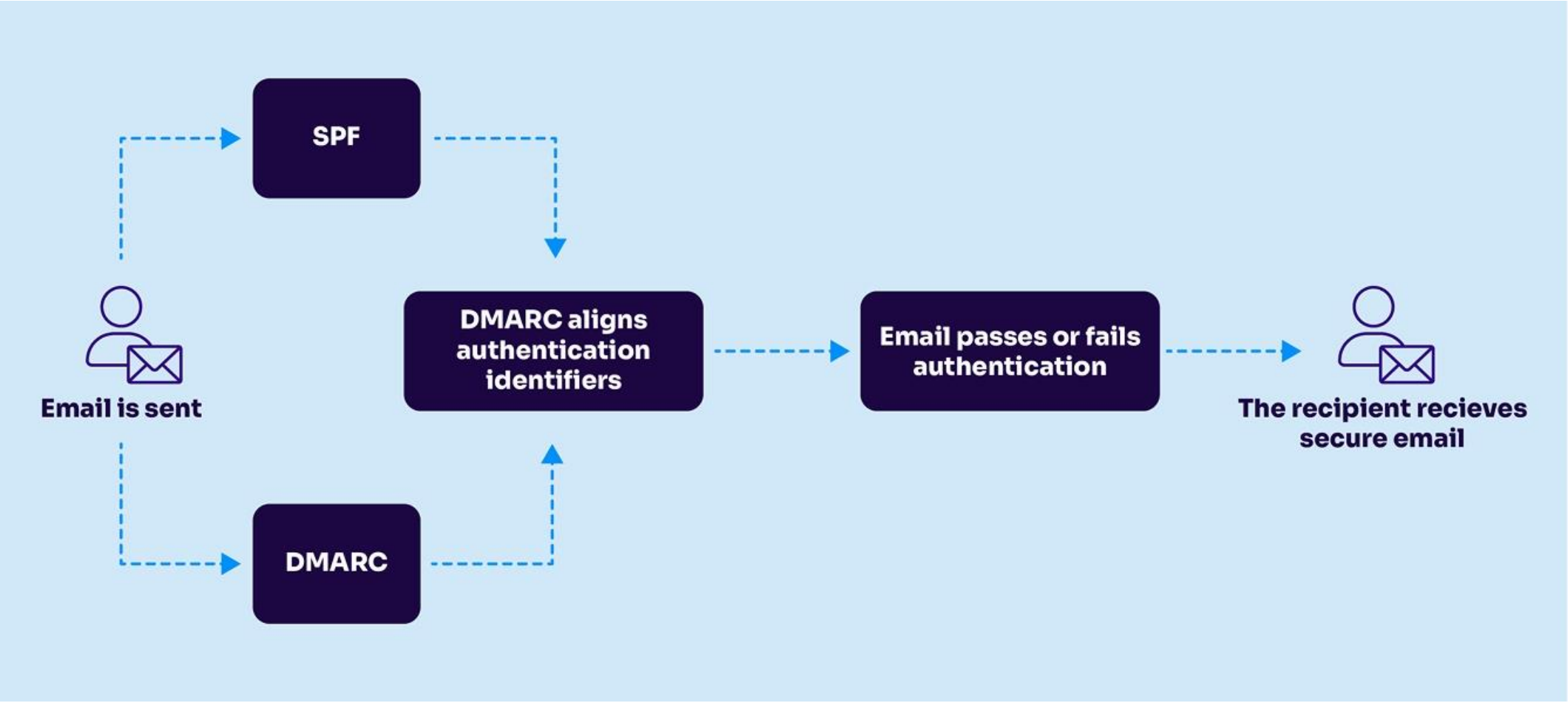
SPF and DKIM are just tools. DMARC is the instruction manual.

> Alignment: Checks if the "From" address matches the SPF/DKIM authenticated domain.

> Reporting: Tells you who is sending mail as you (via daily reports).

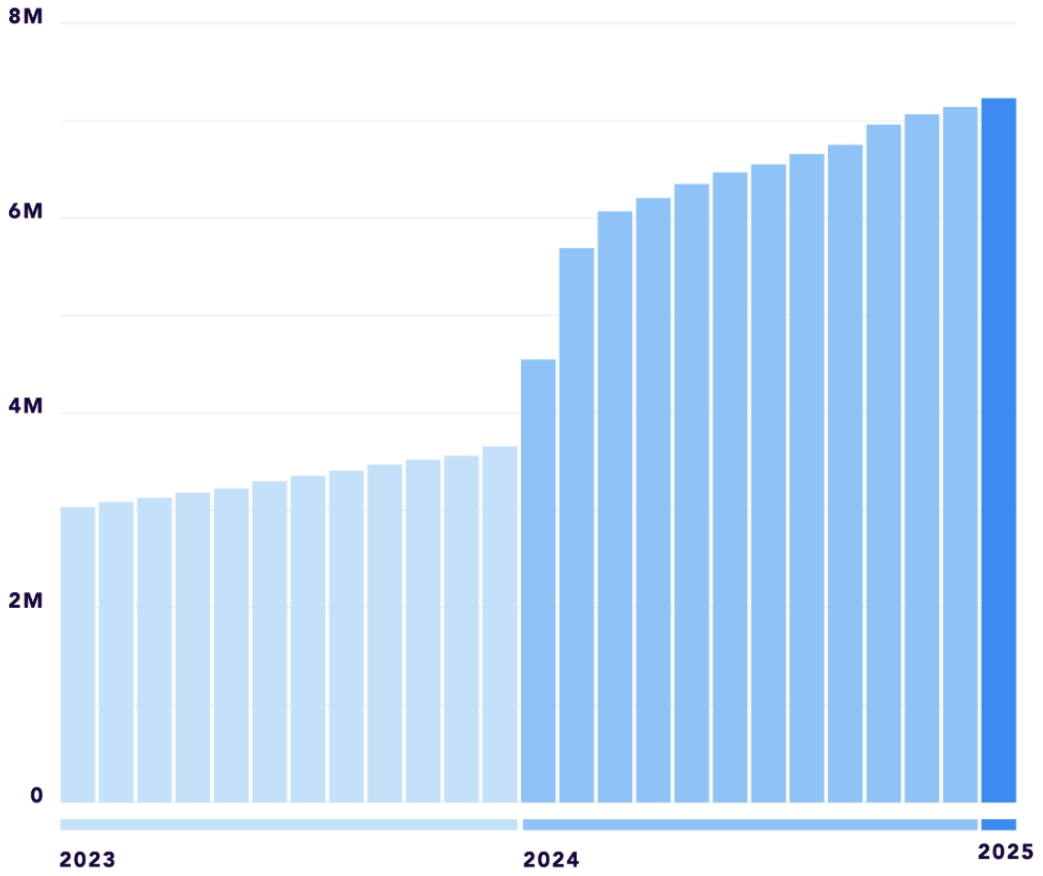
> Enforcement: Tells receivers to Reject, Quarantine, or Do Nothing if checks fail.

DMARC – how it works



src: https://www.valimail.com/dmarc/

DMARC adoption



src:https://www.valimail.com/dmarc/