



Università  
degli Studi  
della Campania  
*Luigi Vanvitelli*

Reti di Calcolatori e Cybersecurity

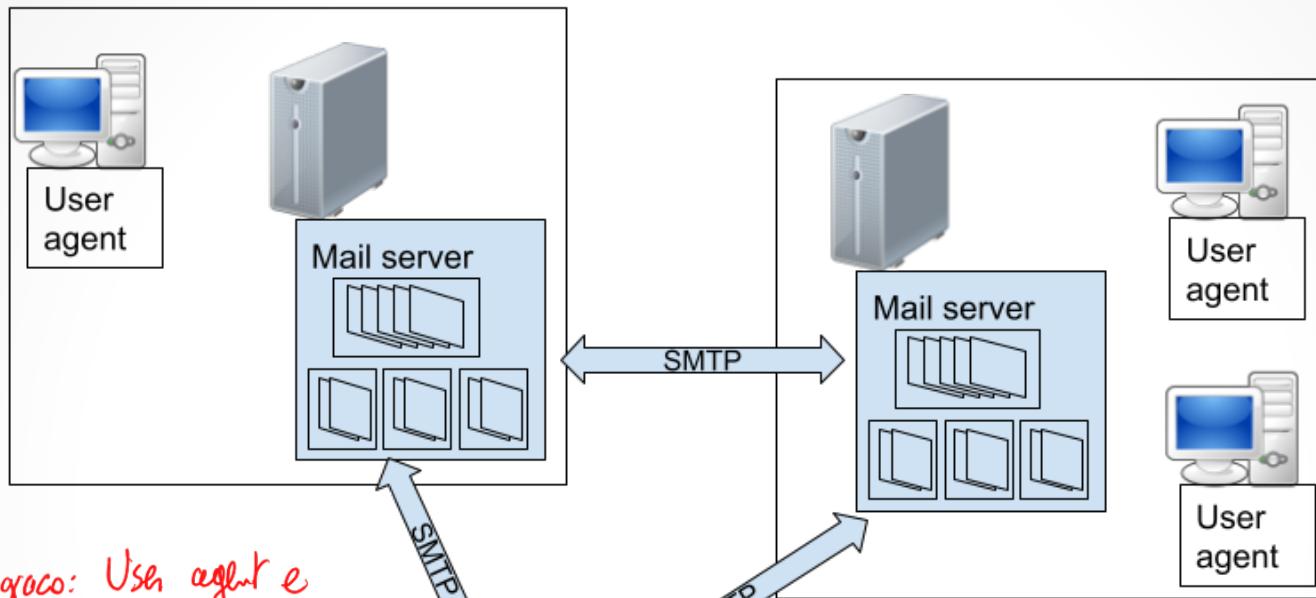
# SMTP – Simple Mail Transfer Protocol

Ing. Vincenzo Abate

# Posta elettronica e Internet

- La posta elettronica è presente fin dagli albori di internet e con il trascorrere del tempo è diventato uno strumento sempre più complesso e potente
- Come il servizio postale ordinario l'email è un mezzo di comunicazione asincrono *Comuniasi non ha bisogno di una fase di setup*
- A differenza del servizio postale ordinario è veloce, facile da distribuire e GRATUITA.
- Sono presenti attualmente molte caratteristiche che rendono la posta elettronica molto versatile: allegati, coll. Ipertestuali, formattazione HTML e foto incorporate.

# Posta elettronica e Internet

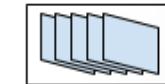
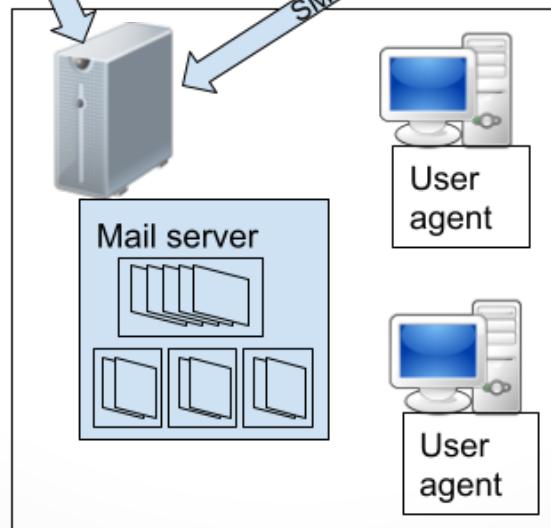


*Emilia in gergo: User agent e mail server*

Ogni mail server ha la sua coda e gestisce mailbox di varie utenti.

Se un file deve mandare da una mailbox alla destinazione,

si usa SMTP.



Coda di messaggi



Mailbox utente

# Posta elettronica e Internet

User Agent (anche detto mail reader) → Client

- composizione, modifica, lettura di messaggi
- es.: Eudora, Outlook, Mozilla Thunderbird, Evolution, Kmail...
- messaggi in uscita ed in entrata immagazzinati sul server
- Il protocollo SMTP viene utilizzato anche tra user-agent e server durante l'invio di una mail



↓  
che permette di scrivere le mail → mail server del mittente lo passa al destinatario

# Posta elettronica e Internet

## Mail server

- mailbox contenente messaggi in entrata (non letti) per l'utente
- coda dei messaggi in uscita contenente i messaggi non ancora recapitati
- protocollo SMTP per l'interazione tra due mail server.
  - “client”: mail server mittente
  - “server”: mail server destinatario
- Un “mail server” funge in momenti diversi da client o da server a seconda del ruolo che ricopre nello scambio del messaggio

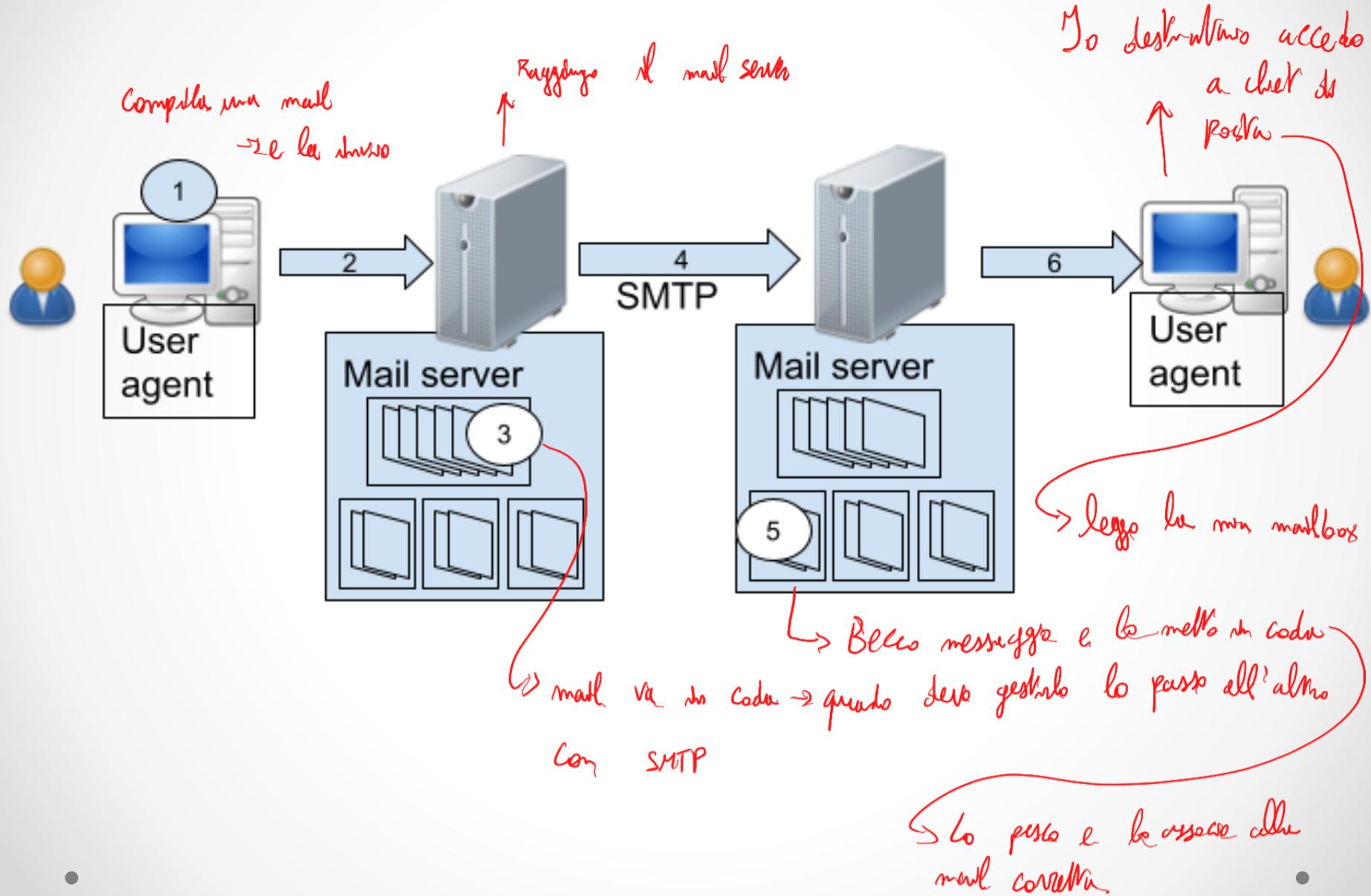
Mail server che manda posta = client

Mail server che riceve = server

# SMTP

- Rappresenta il principale protocollo di livello applicazione per la posta elettronica su internet (RFC 5321)
- Si basa su protocollo TCP (trasferimento dati affidabile) su porto 25
- SMTP presenta un lato client, in esecuzione su mail server e un lato server in esecuzione sul server destinatario:
- Quando un server invia posta ad un altro agisce come client SMTP; quando invece la riceve, agisce da server SMTP

# SMTP



# SMTP

- Trasferimento diretto dal server mittente al server destinatario
- Tre fasi durante il trasferimento via SMTP:
  1. handshaking (“stretta di mano”)
  2. trasferimento del messaggio
  3. chiusura della connessione
- Interazione comando/risposta (command/response)
  - comandi: testo ASCII  
*→ richiede ASCII risposta status code*
  - risposta: codice di stato e descrizione (facoltativa)
- Messaggi codificati con caratteri ASCII a 7-bit

# SMTP

- Usa una connessione persistente *dove chiudere esplicitamente*
- Richiede che il messaggio, comprensivo del contenuto, sia codificato in caratteri ASCII a 7 bit
- Alcune combinazioni di caratteri non sono ammesse (p.es., CRLF.CRLF). Quando queste combinazioni si presentano il messaggio deve essere opportunamente codificato.
- SMTP usa CRLF.CRLF per determinare la fine di un messaggio

# SMTP: interazione c/s

S: 220 hamburger.edu *apre la connessione* ↗ no sono presenti  
C: HELO crepes.fr *chi sono?*

S: 250 Hello crepes.fr, pleased to meet you ↘ handshake

C: MAIL FROM: <alice@crepes.fr> *Ricerca*

S: 250 alice@crepes.fr... Sender ok *il sender esiste sì valicato*

C: RCPT TO: <bob@hamburger.edu> *a chi lo mando?*

S: 250 bob@hamburger.edu ... Recipient ok

C: DATA → *dico cosa gli voglio mandare*

S: 354 Enter mail, end with "." on a line by itself

C: Do you like ketchup?

C: How about pickles?

C: .

S: 250 Message accepted for delivery

C: QUIT

S: 221 hamburger.edu closing connection ↘ *finisce mailer mail e chiedere*

# SMTP: esempio pratico

- `telnet servername 25`
- Si osservi il codice 220 di risposta dal server
- Si inseriscono i comandi HELO, MAIL FROM, RCPT TO, DATA, QUIT
- In questo modo è possibile inviare un'e-mail senza servirsi dello user agent

# SMTP: esempio pratico

comando mail e pw in base 64

Telnet smtp.libero.it

```
220 smtp-36.iol.local smtp-36.iol.local ESMTP server ready
heho pippoppo
250 smtp-36.iol.local hello [193.205.230.117], pleased to meet you
auth login
334 VXNlcm5hbWU6
YWJ...zgz334 UGFzc3dvcmQ6
WW...OTA=235 ... authentication succeeded
mail from: <abatevincenzo83@libero.it> sender ok
rcpt to: <abatevincenzo83@gmail.com> recipient ok
data
354 OK
From: "ENZO" <abatevincenzo83@libero.it>
To: "ENZO" <abatevincenzo83@gmail.com>
Subject: test

Hello,
pippo

.
250 k1EHomvf4CV3k1EyoSfJV mail accepted for delivery
```

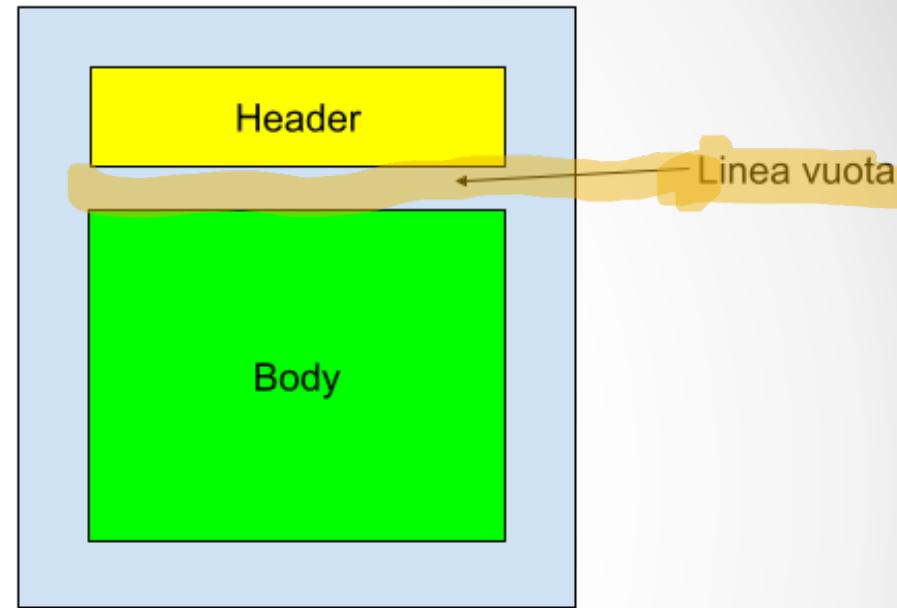
# SMTP: formato messaggi

## Linee di intestazione (header)

- To:
- From:
- Subject:
- differenti da comandi smtp!

## Corpo (body):

- il “messaggio” vero e proprio
- solo caratteri ASCII



# SMTP: estensione MIME

- **MIME**: Multipurpose Internet Mail Extensions, RFC 2045-2056  
*Rende corpo della mail più articolato*
- righe aggiuntive nell'intestazione informano della presenza di un body MIME

versione MIME

metodo utilizzato per codificare i dati

tipo, sottotipo e parametri del contenuto

dati codificati

**From:** alice@crepes.fr  
**To:** bob@hamburger.edu  
**Subject:** Picture of yummy crepe.  
**MIME-Version:** 1.0  
**Content-Transfer-Encoding:** base64  
**Content-Type:** image/jpeg

base64 encoded data ....  
.....  
.....base64 encoded data

# SMTP: tipi MIME

## Text

- sottotipi: plain, html

## Image

- sottotipi: jpeg, gif

## Audio

- sottotipi: basic (8-bit mu-law encoded), 32kadpcm (32 kbps coding)

## Video

- sottotipi: mpeg, quicktime

## Application

- altri dati che devono essere processati da specifiche applicazioni
- sottotipi: msword, octet-stream

# SMTP: esempio multipart

// Example Multipart Email:  
From: sender@example.com  
To: recipient@example.com  
Subject: Multipart Email Example  
Content-Type: multipart/alternative; boundary="boundary-string"

--your-boundary *Separa le varie parti*

Content-Type: text/plain; charset="utf-8"  
Content-Transfer-Encoding: quoted-printable  
Content-Disposition: inline

Plain text email goes here!

This is the fallback if email client does not support HTML

--boundary-string

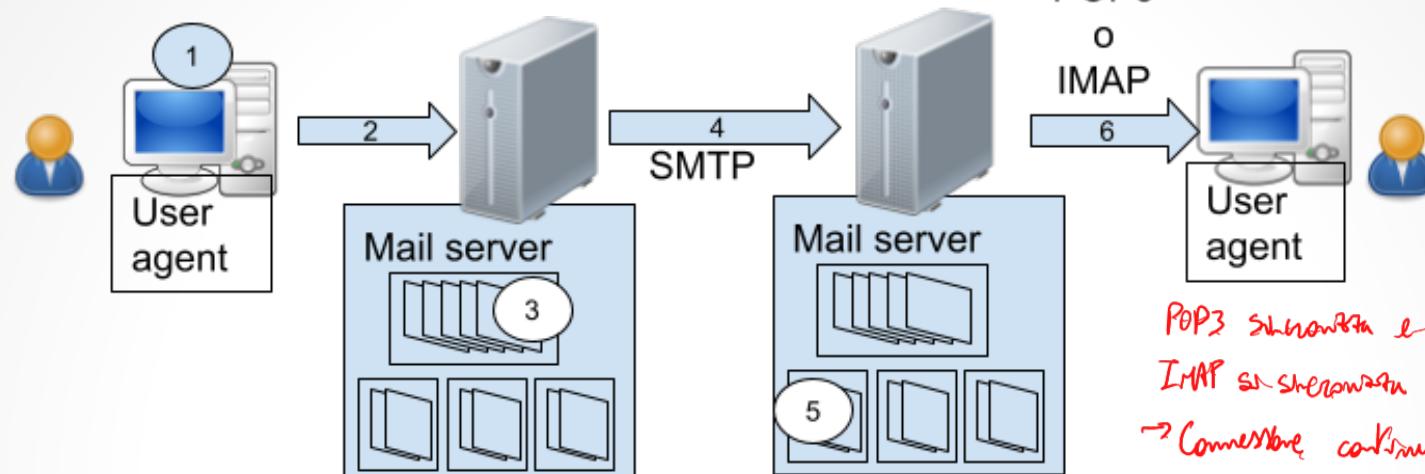
Content-Type: text/html; charset="utf-8"  
Content-Transfer-Encoding: quoted-printable  
Content-Disposition: inline

<h1>This is the HTML Section!</h1>

<p>This is what displays in most modern email clients</p>

--boundary-string--

# Protocolli Accesso alla Posta



SMTP: consegna di messaggi

Protocolli di accesso alla mail: recupero dei messaggi dai server

POP: Post Office Protocol *l'utente scarica la posta dal locale: senza indirizzi posso vedere mail, autorizzazione (agent ↔ server) e download* ma salvo velocemente

IMAP: Internet Mail Access Protocol [RFC 2060] *indirizzo → user loc*

più complicato e potente

manipolazione avanzata dei messaggi sul server

HTTP: gmail, Hotmail, Yahoo! Mail, ecc. *Protocollo di accesso con HTTP con web app*

# SMTP: esempio POP3

## Autorizzazione

### comandi del client:

- user: **specifica la username**
- pass: **specifica la password**
- il server risponde  
**+OK -ERR**

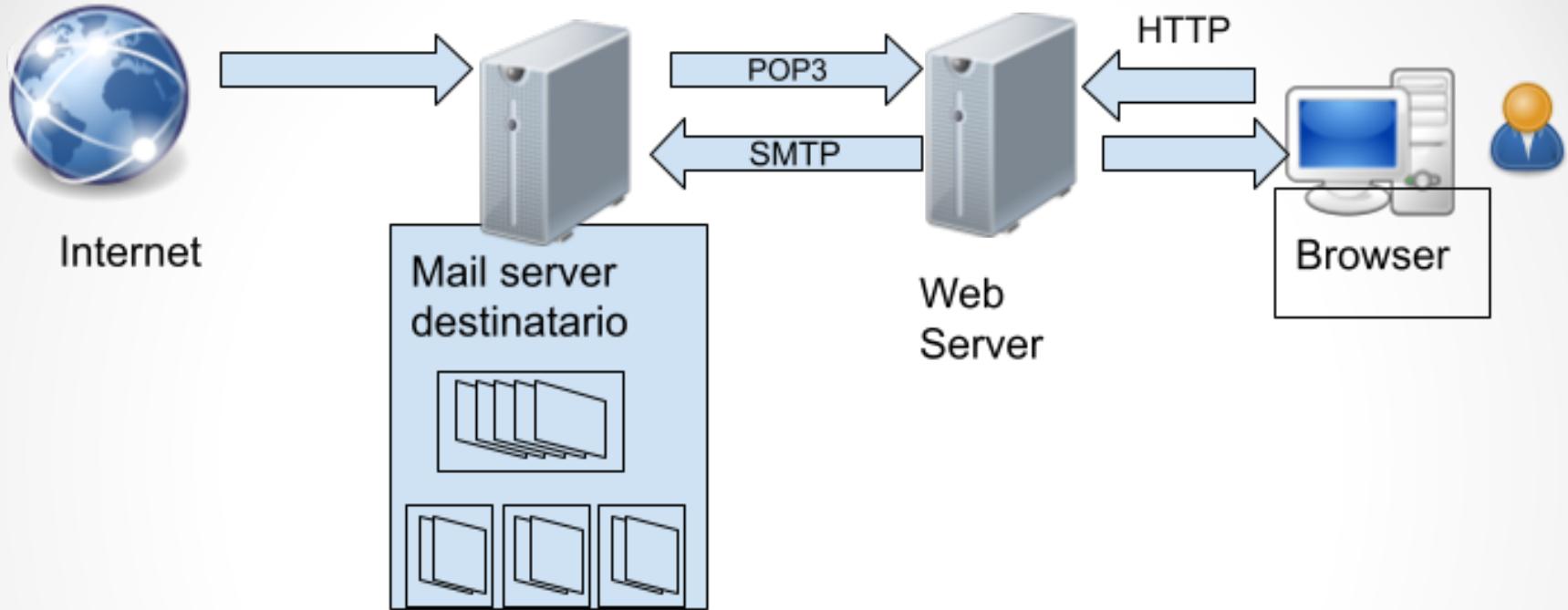
## Fase di scambio

### comandi del client:

- **list: visualizza la lista dei messaggi**
- **retr: preleva il messaggio per numero**
- **dele: elimina il messaggio dal server**
- **quit: chiude la sessione**

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 2 contents>  
S: .  
C: dele 2  
C: quit  
S: +OK POP3 server signing off

# Accesso via web



- Molti siti web forniscono accesso alle proprie caselle di posta (gmail, hotmail, Yahoo!, ecc.)
- In questo caso non serve avere uno user agent installato e correttamente configurato per ricevere ed inviare posta.
- È sufficiente disporre di un qualsiasi browser
-