Electronic Mails

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

❑ User Agent

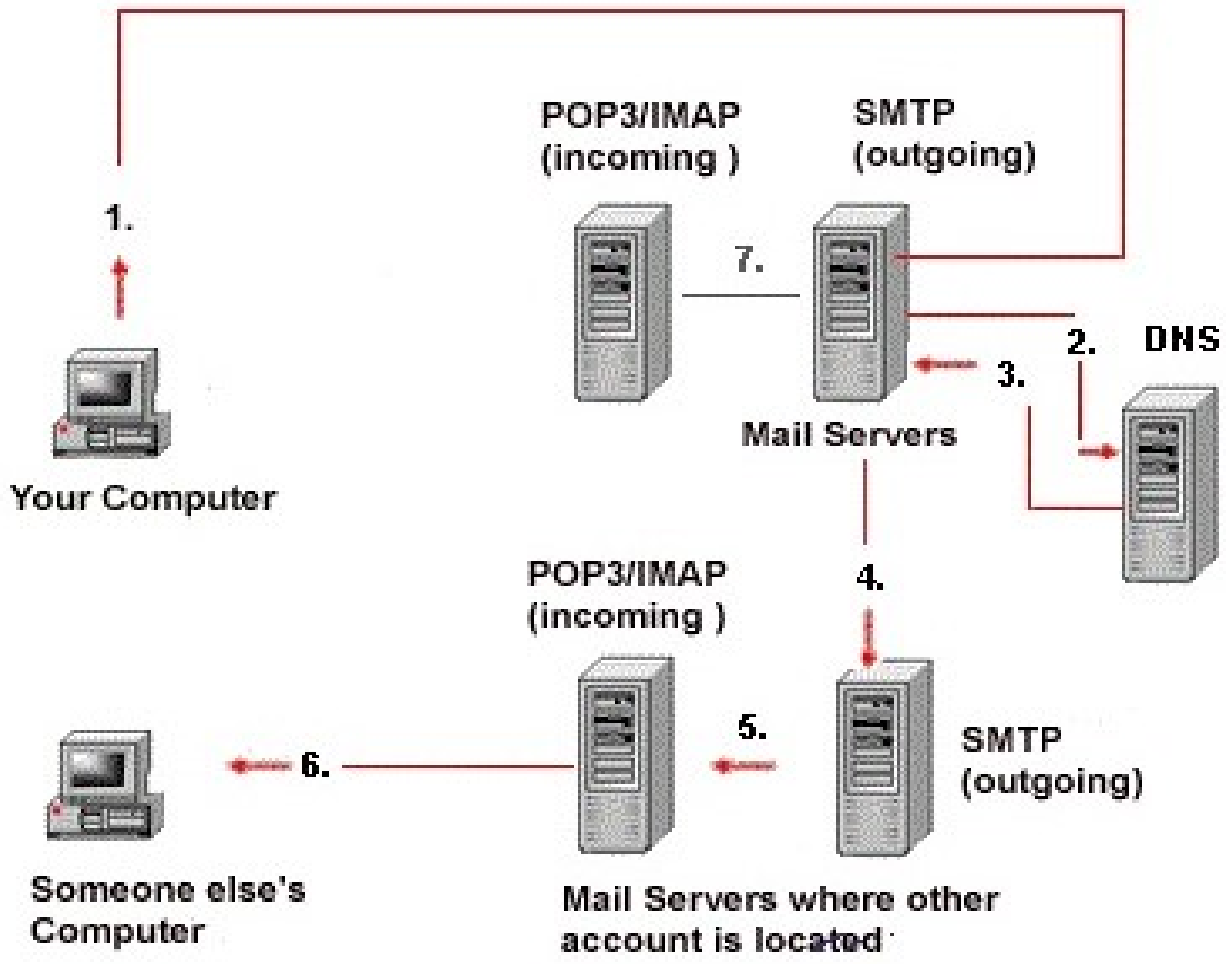
a.k.a. "mail reader", client-based-MTA

➢ composing, editing, reading mail messages

e.g.- Outlook, elm, Netscape Messenger

outgoing, incoming messages stored on server

How does email work?



SMTP Programming

import ***smtplib***

s = smtplib.SMTP(***host***=*' '*, ***port***= , ***local_hostname*** = *None*)

❖ ***host*** – This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.

❖ ***port*** – If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.

❖ ***local_hostname*** – If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.

smtp*lib* methods

s.starttls()

❖ Put the SMTP connection in ***TLS*** (Transport Layer Security) mode. All SMTP commands that follow will be encrypted.

s.login (***user***, ***password*** **)**

❖ Log in on an SMTP server that requires authentication. The arguments are the username and the password to authenticate.

s.sendmail(*from_addr, to_addrs, msg* **)**

❖ The required arguments are an **from-address** string, a list of **to-address** strings (a bare string will be treated as a list with 1 address), and a message string.

Ex :

to_addrs **=** **‘**exp1@gmail.com**’** **OR**

to_addrs **= [** **‘**exp1@gmail.com**’** **,** **‘**exp2@gmail.com**’** **, …]**

NOTE :

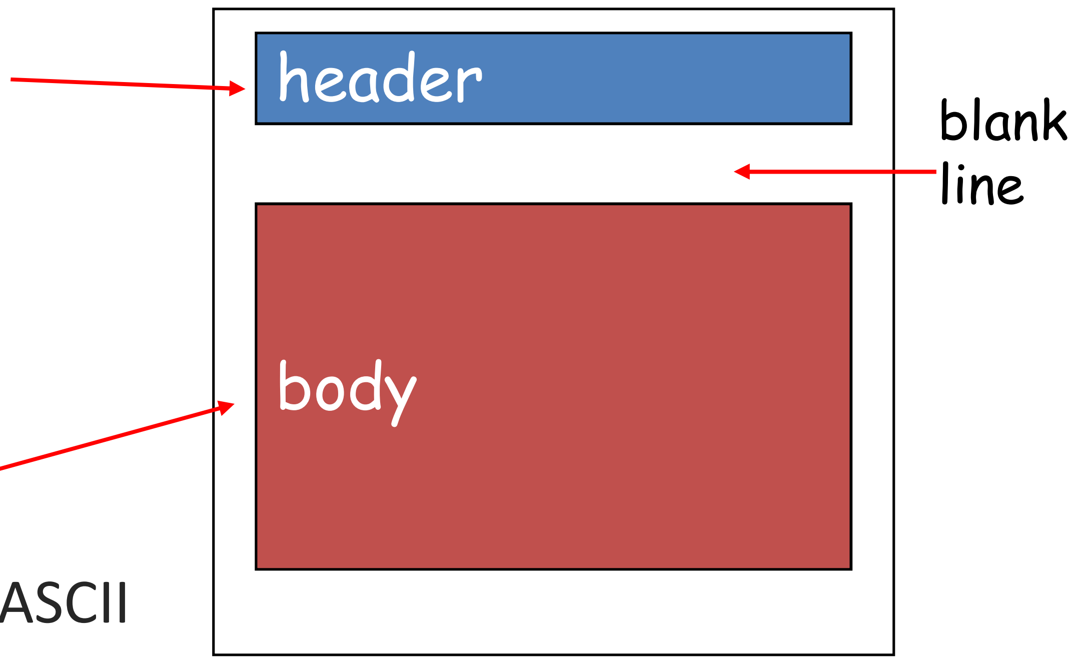
The ***from_addr*** and ***to_addrs*** parameters are used to construct the message envelope used by the transport agents. sendmail does not modify the message headers in any way. So we have to arrange header by our own manually or with help of another library as example : ***email.message***

import ***email.message***

Mail message format

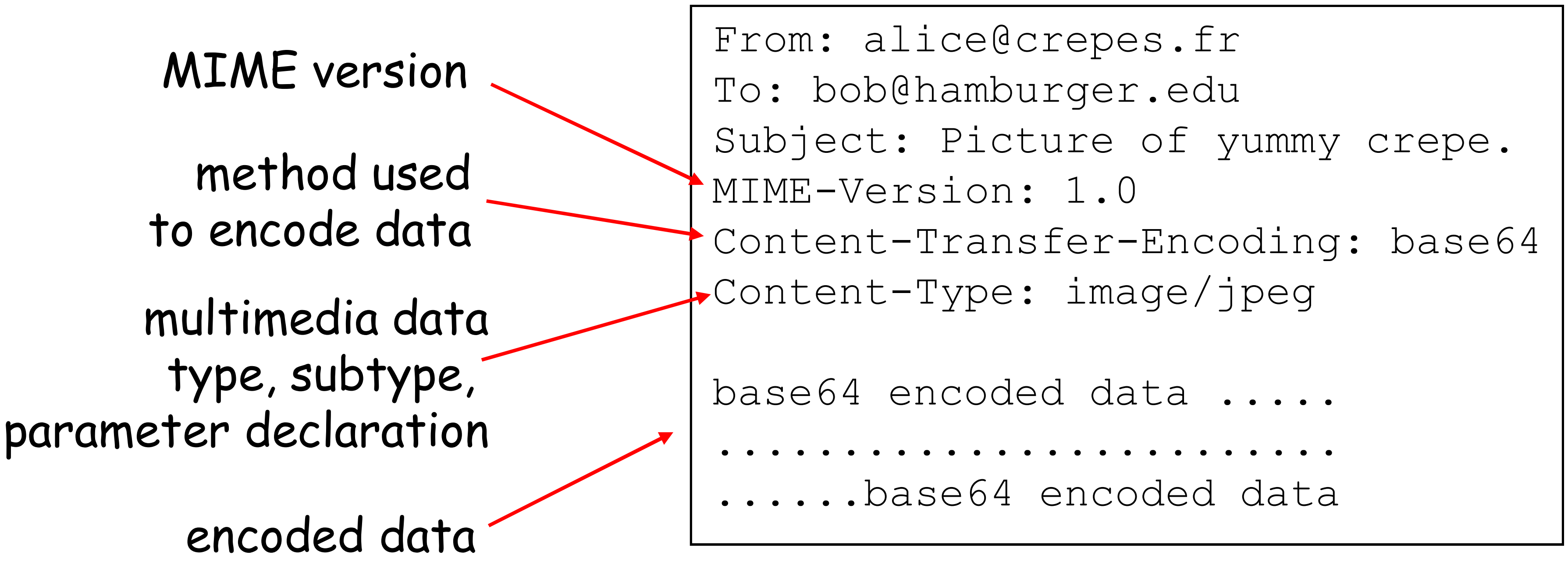
standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:
- body
  - the "message", ASCII characters only



Message format: multimedia extensions

- **MIME :** Multipurpose Internet Mail Extension
  - additional lines in msg header declare MIME content type


MIME version
method used to encode data
multimedia data type, subtype, parameter declaration
encoded data

```
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
```

import ***email.message***

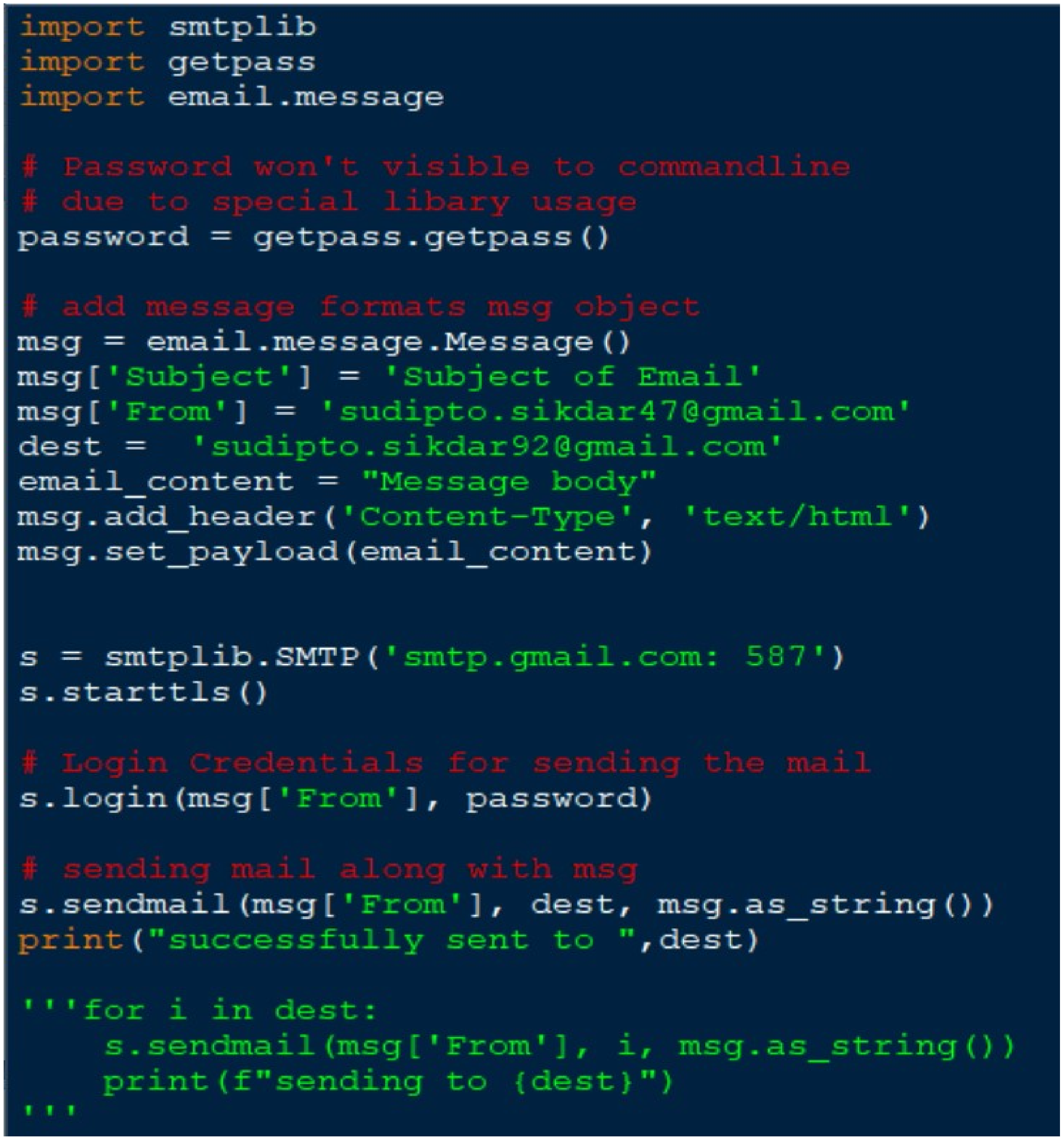
❖ This python library provides the core functionality for setting and querying header fields, for accessing message bodies, and for creating or modifying structured messages.

msg **=** email.message.**Message***()*

❖ This rules it specifies to update and serialize the representation of the message. which follows the rules of the email for line endings.

msg.as_string()

❖ Return the entire message flattened as a string. Flattening the message need to be filled in to complete the transformation to a string.



```python
import smtplib
import getpass
import email.message

# Password won't visible to commandline
# due to special libary usage
password = getpass.getpass()

# add message formats msg object
msg = email.message.Message()
msg['Subject'] = 'Subject of Email'
msg['From'] = 'sudipto.sikdar47@gmail.com'
dest =  'sudipto.sikdar92@gmail.com'
email_content = "Message body"
msg.add_header('Content-Type', 'text/html')
msg.set_payload(email_content)


s = smtplib.SMTP('smtp.gmail.com: 587')
s.starttls()

# Login Credentials for sending the mail
s.login(msg['From'], password)

# sending mail along with msg
s.sendmail(msg['From'], dest, msg.as_string())
print("successfully sent to ",dest)

'''for i in dest:
    s.sendmail(msg['From'], i, msg.as_string())
    print(f"sending to {dest}")
'''
```