Le Mail :
Architecture et protocoles

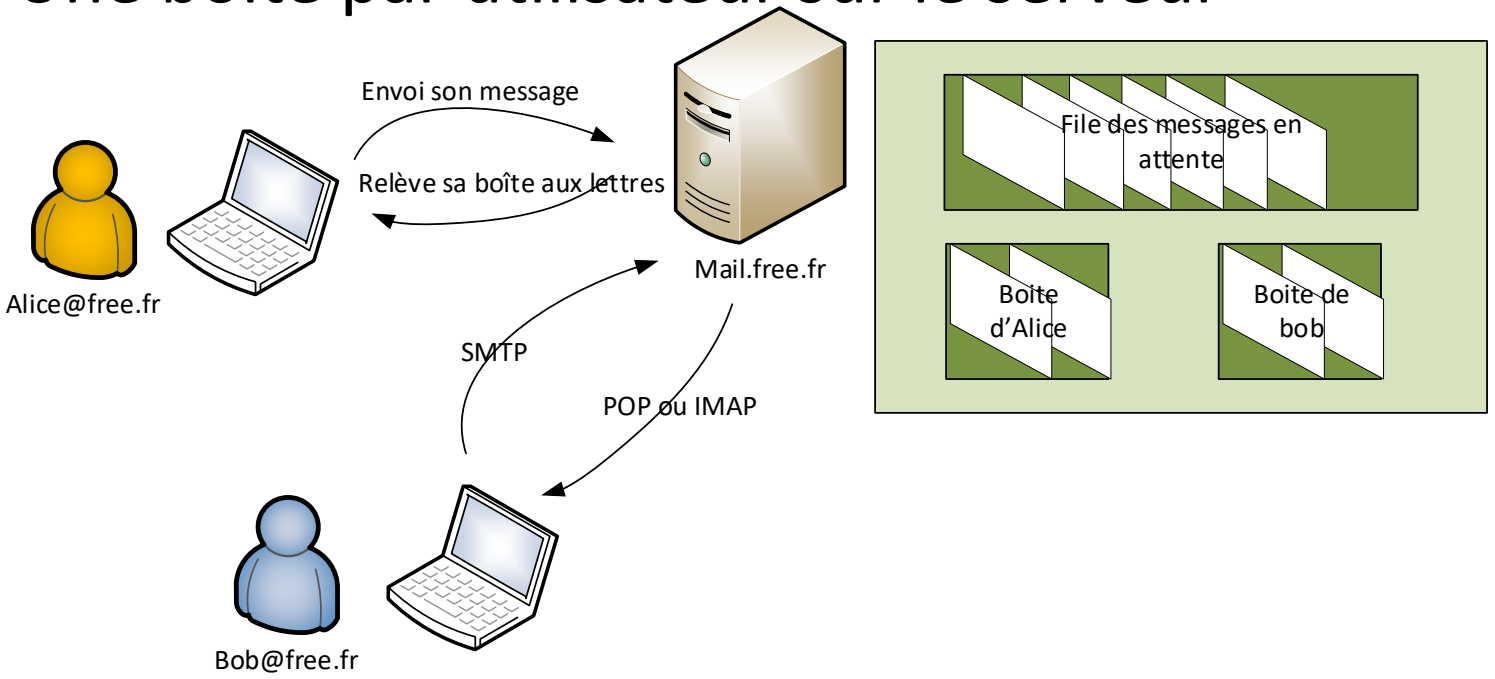
M Berthet

Service de courrier (mail)

- Prévu initialement pour échanger des messages simples en texte ASCII, il permet aujourd'hui de joindre n'importe quelle pièce jointe. On peut donc l'utiliser également pour s'échanger des fichiers.
- La taille des pièces jointes est limitée par les serveurs par lesquels le message transite.
- Plusieurs protocoles sont impliqués :
  - SMTP port 25
  - POP3 port 110
  - IMAP port 143
- Et autant pour les versions sécurisés (chiffrement):
  - Secure SMTP (SSMTP) - port 465
  - Secure IMAP (IMAP4-SSL) - port 585
  - IMAP4 over SSL (IMAPS) - port 993
  - Secure POP3 (SSL-POP) - port 995
- Rôle des ports TCP/UDP ?

Organisation du mail

- Cas 1 : Alice et Bob ont un compte mail chez free.fr et utilisent un client de messagerie sur leur ordinateur ou leur téléphone
- Un serveur de mail pour le domaine free.fr
- Une boîte par utilisateur sur le serveur

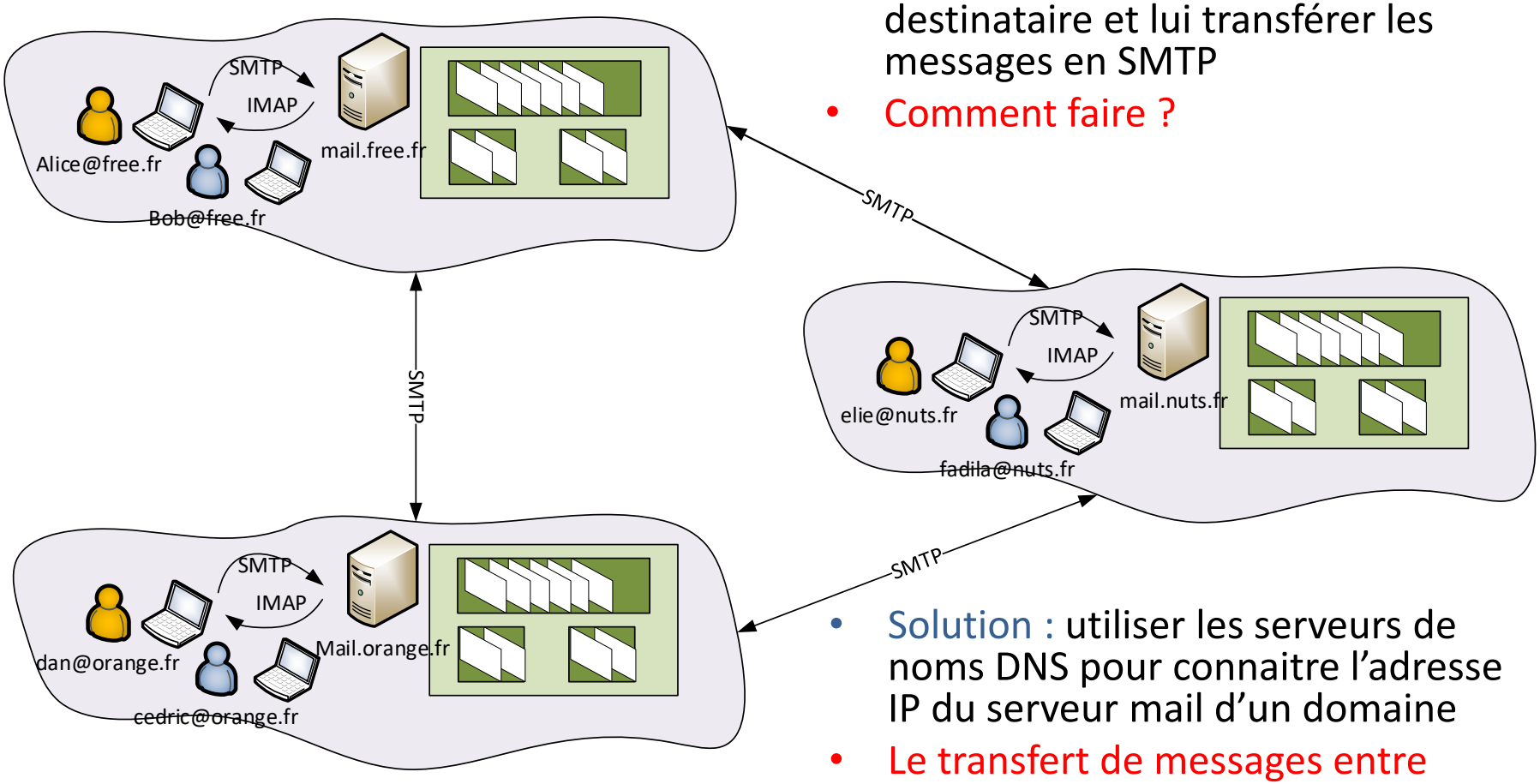


Rôles du serveur

- Etre toujours disponible pour réceptionner de nouveaux messages, et permettre aux utilisateurs de relever leurs mails à tout moment
- Gérer les utilisateurs : comptes, mots de passe, chiffrement des transferts
- Gérer les requêtes multiples, en les organisant et en les traitant à tour de rôle.
- Conserver des copies des messages et organiser les boîtes en dossiers, sous dossiers, marquage des messages ..

Cas 2 : Différents domaines

- Nouveau rôle du serveur, localiser le serveur de messagerie du destinataire et lui transférer les messages en SMTP
- Comment faire ?

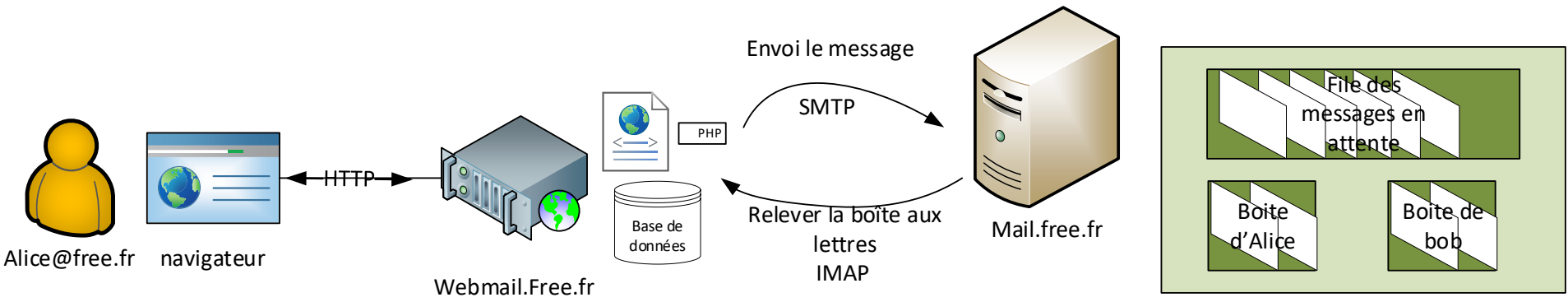


- Solution : utiliser les serveurs de noms DNS pour connaitre l'adresse IP du serveur mail d'un domaine
- Le transfert de messages entre serveurs se fait sans authentification

Client de messagerie

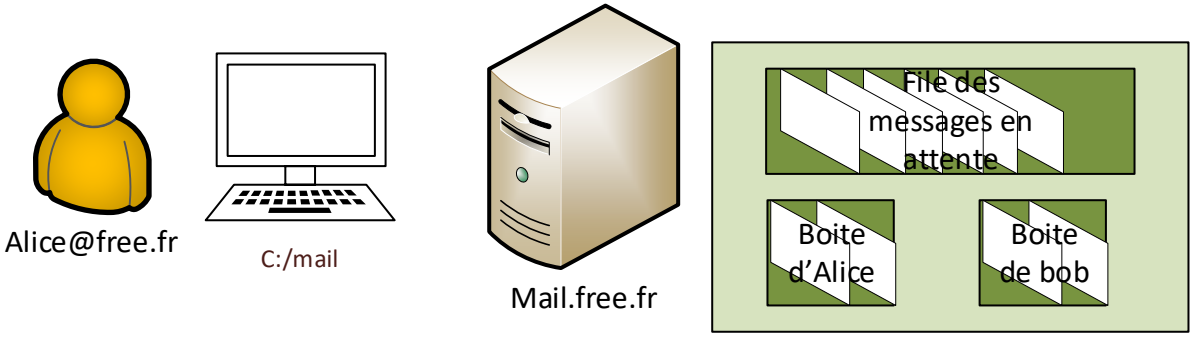
- C’est un programme pour lire et écrire ses messages sur un ordinateur, ou un tel mobile
- Permet de garder une copie des messages et de pouvoir les lire ensuite, déconnecté du réseau Internet
- Doit être configuré, avec le nom du serveur d’envoi (SMTP), le nom du serveur où se trouve la boîte aux lettres (POP, IMAP) et les ports pour s’y connecter.
- Exemples : courrier sous Windows, mail sous OsX, Thunderbird partout, appli gmail sur Android …

Cas 3 : Webmail



- Alice utilise un navigateur, elle doit connaître l'URL de son webmail, son adresse mail et son mot de passe pour envoyer et recevoir, depuis n'importe où ?
  - + pas de configuration, mail accessible sur n'importe quel navigateur
  - - connexion Internet nécessaire pour relire les messages
- Architecture qui nécessite un serveur Web/php/bd pour faire l'interface avec le système de mail. Ce sont donc des scripts PHP (ou autre langage serveur) qui font le dialogue avec le serveur de mail en SMTP/IMAP

Cas 4: session sur le serveur



- Alice dispose d'une session sur le serveur de mail.
- Elle utilise la commande mail pour écrire ou lire ses messages.
- Sa boîte mail est un dossier dans son home directory
- Elle peut aussi accéder à distance en SSH (secure shell)
- Peu courant pour les utilisateurs lambda, très utile pour les administrateurs systèmes

Format des messages

- Dans la version SMTP de base, les messages doivent être codés en ASCII pour le transfert
- Pas de problème pour le texte simple.
- Pour les autres contenus : texte évolué, images, sons, fichiers binaires de tout type, le message doit être encodé en ASCII, avant l'envoi et décodé à la réception.
- Les types MIME (*Multipurpose Internet Mail Extensions*) permettent d'identifier le contenu du message.
- Un message avec trois pièces jointes est donc transféré sous la forme d'un seul message encodé en ASCII.
- Les première lignes d'un message constituent l'en tête, sous la forme *variable : valeur*
- Le corps de message contient le mail et les pièces jointes

Structure des messages SMTP

- Le protocole SMTP spécifie :
  - le format des adresses des utilisateurs suivant une notation Internet classique faisant figurer le nom de l'utilisateur suivi du nom de domaine (asterix@babaorum.fr par exemple) .
  - les champs des courriers (*from , to*…) ;
  - les possibilités d'envoi groupé :
    - Le champ CC pour *Carbone Copy* ou Copie Conforme permet d'envoyer le même message à plusieurs personnes, tous les noms des destinataires apparaîtront en clair dans le champ *to* ;
    - Le champ BCC (*Blinde Carbone Copy*) ou CCI (Copie Conforme Invisible) permet de masquer le nom des autres destinataires dans le champ *to* ;
- la gestion des heures ;
- le codage utilisé pour le message et les fichiers joints (« attachés ») :
  - texte pur codé en ASCII 7 (RFC 822) ou 8 bits pour une prise en compte des caractères accentués ;
  - standard MIME (*Multipurpose Internet Mail Extension*) pour du texte formaté, des images ou du son.

Exemple de code source d'un message

Lignes d'en tête

Message Texte Encodage 8 bits

Pièce jointe Encodage Base 64

```
To: "matthieu berthet" <berthet@univ-eiffel.fr>
Subject: =?iso-8859-1?Q?TD_r=E9seaux=2C_suite?=
Date: Tue, 29 Jan 2022 16:20:48 +0100
Message-ID: <AFEOLMLMMBNPJJBBPLPNOEDECHAA.caporali.stephane@neuf.fr>
MIME-Version: 1.0
Content-Type: multipart/mixed;
----=_NextPart_000_002A_01C86292.E8677260
This is a multi-part message in MIME format. ------=_NextPart_000_002A_01C86292.E8677260
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 8bit
Voici les deux prochaines TD de réseaux sur le thème de la sécurité. Le premier a été un peu abordé hier, la correction des
    exams de réseau et signaux ayant pris plus de temps que prévu. Nous utiliserons une salle de TP pour que les élèves
    s'aident du réseau internet. A bientot Stéphane
……=_NextPart_000_002A_01C86292.E8677260
Content-Type: application/pdf;
name="=?iso-8859-1?Q?Microsoft_Word_-_TD_1_r=E9partition_de_charges.pdf?="
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="=?iso-8859-1?Q?Microsoft_Word_-_TD_1_r=E9partition_de_charges.pdf?=«
 JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0ZURlY29k
    ZT4+CnN0cmVhbQp4nN1aS48UNxC+z6/oW3oijeP3g0ukhChSFEUB5gY5DLuzZNHuwi67COVn8As5
    woXfkPKj29Vtex4EBApIrNd2lcv1/FzNdUcJ4x31f4fByeXih4eme/ZqEaa7h7+mwc2zxfXCEuH/
    hAk8PrnsfloDoeuAzfpsQYlzTrKwxjrODaGyE5ZIS3W3vlw87h8tVxIYMGb6h8uVIMIppfuf85Av…….etc
```

En têtes SMTP

- Le tableau suivant donne les principaux champs d'en-tête RFC 822 liés au transport du message (to, from…) et ses caractéristiques (date, sujet…).

En-tête	Signification
To :	Adresse(s) électronique(s) du(des) destinataire(s) primaire(s)
Cc :	Adresse(s) électronique(s) du(des) destinataire(s) secondaire(s)
Bcc :	Adresse(s) électronique(s) du(des) destinataire(s) en copie caché
From :	Adresse électronique de l'auteur du message
Received	Adresse électronique de l'expéditeur du message
Return Path :	Peut être utilisé pour identifier un chemin de retour vers l'émetteur
Date :	Date et heure de l'envoi du message
Reply to :	Adresse électronique du destinataire d'une réponse
Message id :	Numéro servant à référencer le message
In Reply to :	Numéro (message id) du message auquel on répond
Subject :	Résumé en une ligne du message

SMTP et MIME

- Le standard MIME (RFC1521) reprend le standard simple de codage ASCII (RFC 822) mais en structurant le corps du message et en définissant des règles de codage pour le corps du message et les fichiers joints :
  - un codage base64 pour les messages binaires (groupes de 24 bits segmentés en 6 bits et ASCII légal : A pour 0, B pour 1…) voir la table ; 3 mots de 8 bits donnent 4 mots de 6 bits et 2bits à 00 pour chaque mot, le volume augmente donc d'un tiers
  - un codage QP (*Quoted Printable* – codage ASCII sur 7 bits et une séquence spécifique composé du signe égal et de la valeur du caractère pour les codes supérieurs à 127) pour les messages texte ; (ex =E9 pour é)
  - un codage spécifique lié à l'application (son, vidéo…).
- Pour définir ces nouveaux codages, MIME utilise d'autres en-têtes spécifiques, par exemple :
  - « Content-type : text:html » précise que le texte du message est codé en langage html ;
  - « Content-type : video:mpeg » donne le format d'un fichier vidéo attaché…

Exercice 1

Code source en tête de message :

```
Return-Path: <mlegas@hotmail.fr>
Delivered-To: berthet@iut-velizy.uvsq.fr
Received: from localhost (unknown [127.0.0.1])
        by horus.iut-velizy.uvsq.fr (Postfix) with ESMTP id E1F5339800
        for <berthet@iut-velizy.uvsq.fr>; Tue, 26 Aug 2012 11:19:06 +0200 (CEST)
Received: from horus.iut-velizy.uvsq.fr ([127.0.0.1])
        by localhost (horus.iut-velizy.uvsq.fr [127.0.0.1]) (amavisd-new, port 10024)
        with ESMTP id uctqB4e33R6B for <berthet@iut-velizy.uvsq.fr>;
        Tue, 26 Aug 2012 11:18:41 +0200 (CEST)
Received: from soleil.uvsq.fr (soleil.uvsq.fr [193.51.24.1])
        by horus.iut-velizy.uvsq.fr (Postfix) with ESMTP id BCE9239804
        for <berthet@iut-velizy.uvsq.fr>; Tue, 26 Aug 2012 11:18:41 +0200 (CEST)
Received: from bay0-omc2-s6.bay0.hotmail.com (bay0-omc2-s6.bay0.hotmail.com
[65.54.246.142])by soleil.uvsq.fr (8.14.1/jtpda-5.4) with ESMTP id m7Q9IWmj059857
          for <berthet@iut-velizy.uvsq.fr>; Tue, 26 Aug 2012 11:18:38 +0200 (CEST)
Received: from BAY130-W1 ([65.55.135.36]) by bay0-omc2-s6.bay0.hotmail.com with
Microsoft SMTPSVC(6.0.3790.3959); Tue, 26 Aug 2012 01:47:09 -0700
Message-ID: <BAY130-W196A51BC81C645318122CAB660@phx.gbl>
Content-Type: multipart/alternative; boundary="_8dec2f80-8523-462c-9a33-
810edf2371bd_"
From: =?iso-8859-1?Q?M=E9lanie_Legas?= <mlegas@hotmail.fr>
To: <berthet@iut-velizy.uvsq.fr>
Subject: =?iso-8859-1?Q?FW:_Rattrapage_r=E9seaux?=
Date: Tue, 26 Aug 2012 10:47:09 +0200
MIME-Version: 1.0
```

Questions :

1. Qui est l'émetteur de ce message ?
2. Qui est le destinataire ?
3. Une ligne avec la variable Received est ajoutée au message, au passage de chaque serveur SMTP. Par combien de serveurs SMTP le message est-il passé pour atteindre le serveur du destinataire ?
4. Identifiez les différents serveurs (noms et adresses IP) dans l'ordre.
5. Pourquoi y a-t-il plusieurs serveurs de mail pour le même domaine ?
6. Combien de temps a mis le message pour arriver à destination ?

Exercice 2 :

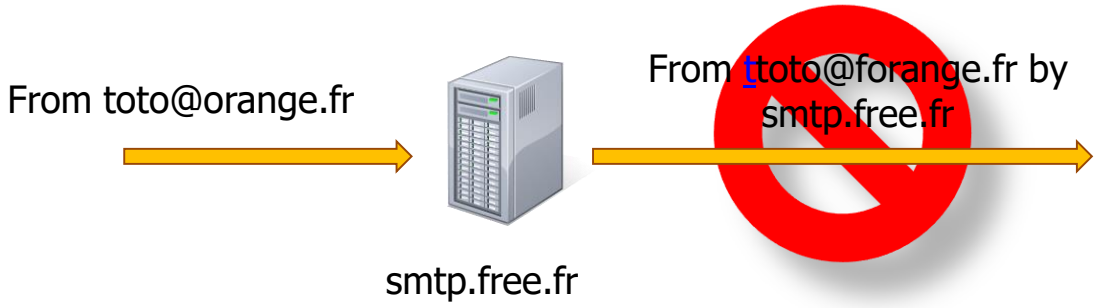
Code source complet d'un message (voir pdf sur elearning)

1. Qui est l'émetteur du message ?
2. Qui est le destinataire ?
3. Par combien de serveurs de mail le message passe-t-il ?
4. En combien de temps ?
5. Tous les serveurs utilisent ils le protocole SMTP ?
6. Sur quels transferts le protocole SSL est il utilisé ?
7. Le message est il en clair dans les serveurs ?
8. Que contient le message ?
9. Quel est le rôle de l'encodage base64 ?
10. Le message est il lisible ?

Relayage SMTP

- Le « relayage » SMTP (relaying) permet à un serveur de mail de transférer un message au serveur du destinataire sans s'identifier.
- Le relayage entre serveurs est donc toujours ouvert

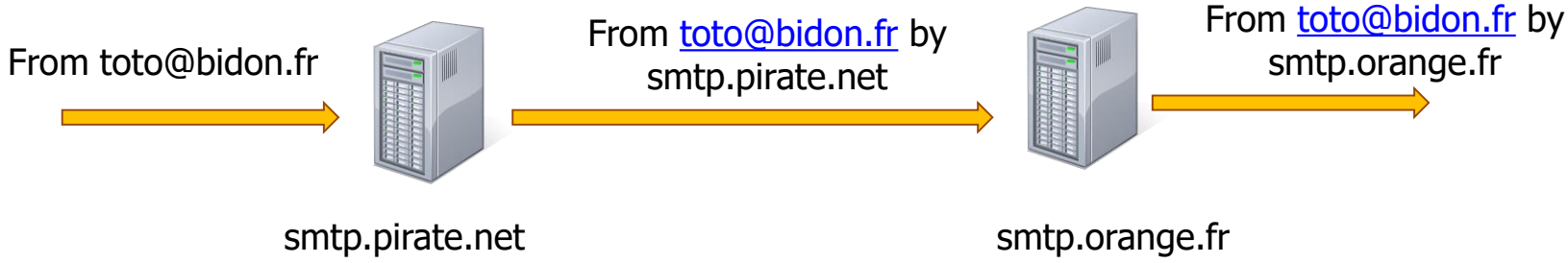
- SMTP aussi permet à un abonné d'envoyer du courrier en utilisant un serveur de mail, si le relayage des clients est ouvert, alors il peut utiliser un autre serveur de mail que celui de son fournisseur d'accès
- En pratique le relayage des clients est fermé chez la plupart des FAI. La vérification se fait à partir de l'adresse mail et de l'adresse IP du client



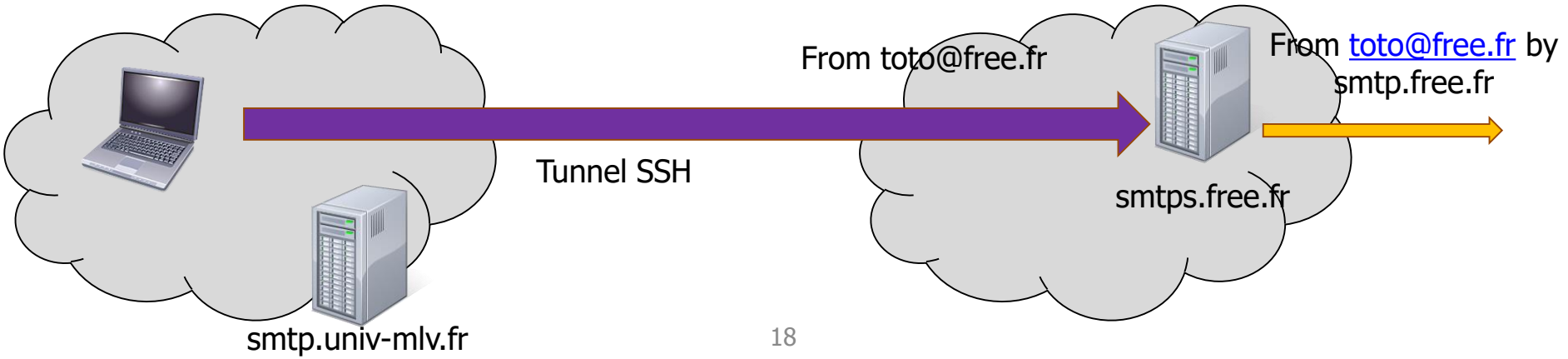
- Le client doit donc modifier la configuration de son client de messagerie lorsqu'il change de réseau pour utiliser le serveur SMTP de son domaine, ou bien utiliser SMTPS, qui permet l'authentification du client

SPAM et relayage ouvert

- Un pirate utilisant un serveur SMTP ouvert pour les clients, peut facilement envoyer des SPAM avec des adresses « bidons »



- Pour pouvoir utiliser votre serveur de mail depuis n'importe quel réseau vous devez utiliser un tunnel chiffré (SMTPS), ainsi vous vous authentifiez sur le serveur.



Protocole SMTP

- SMTP (*Simple Mail Transport Protocol*) : protocole de gestion du courrier électronique sur Internet.
- C'est un protocole client serveur
- Les serveurs sont chargés du stockage dans des boites aux lettres privées (BAL) et du transport du courrier.
- SMTP permet de lire et d'envoyer du courrier lorsque l'on est logué sur le serveur.
- SMTP permet d'envoyer du courrier à partir d'une machine distante.
- SMTP permet à deux serveurs de s'envoyer du courrier. Chacun utilise un client et un serveur

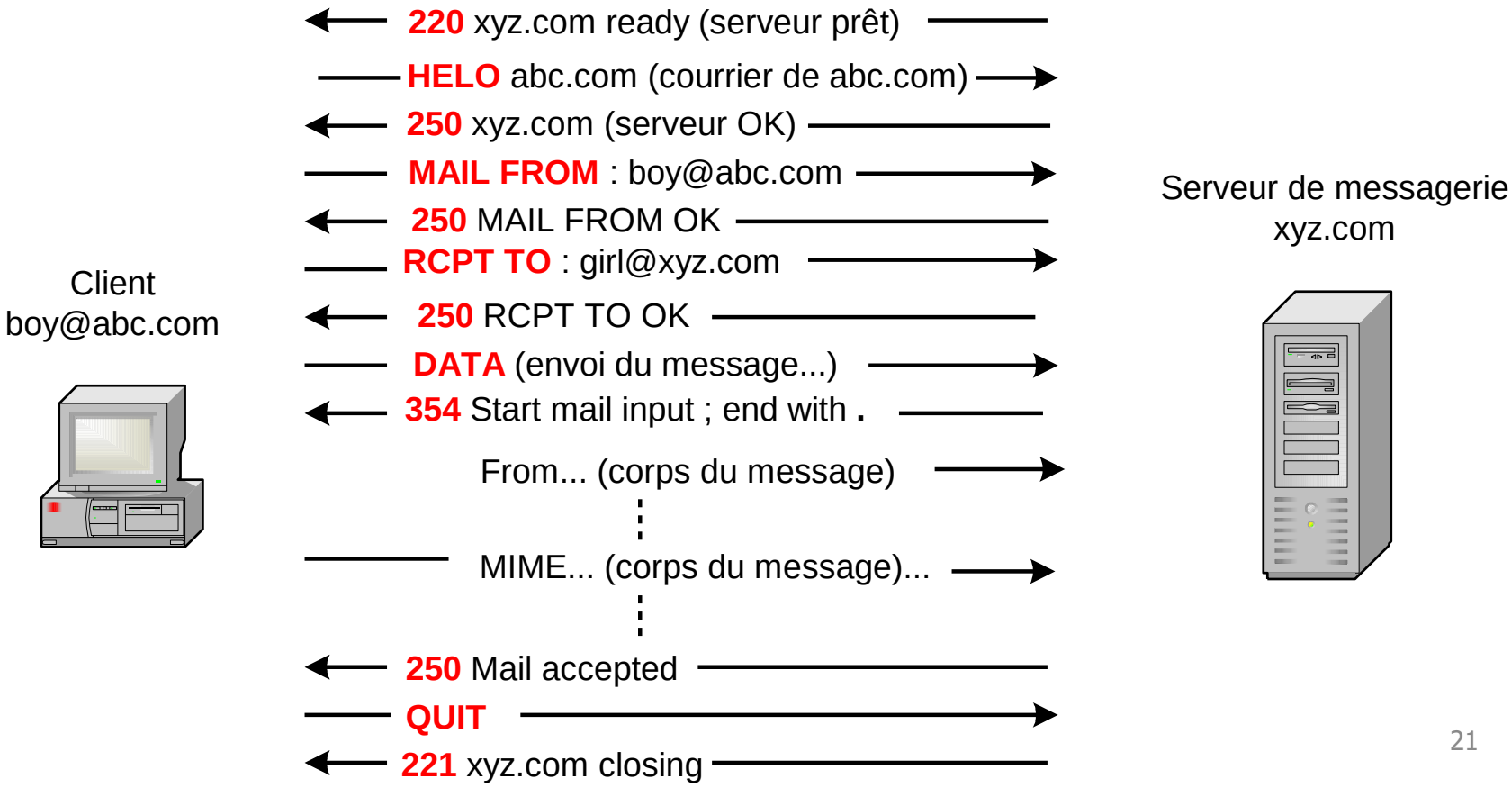
Commandes SMTP

- Une fois formatés, les messages sont envoyés en utilisant les commandes SMTP :
  - Commandes d'envoi constituées de quatre lettres ;
  - Commandes de réponse du serveur constituées d'un code sur trois chiffres suivi d'un message texte.

Commandes d'envoi	**Fonction**
HELO exp	Requête de connexion en provenance d'un expéditeur SMTP.
MAIL FROM ; adr_exp	Adresse de l'expéditeur, annonce le début d'un échange.
RCPT TO :ad_dest	Spécifie un destinataire, la commande peut être répétée.
DATA	Le récepteur interprète toutes les données suivantes comme faisant partie du message SMTP jusqu'à l'apparition d'un point correspondant à 2 sauts de ligne (CR LF CR LF).
QUIT	Demande au récepteur d'envoyer la réponse OK et de refermer la connexion.
Commandes de réponse	
250	Action demandée correctement effectuée (message OK).
354	Le message peur être transmis.
451	Action demandée annulée : erreur pendant le traitement.
550	Action non effectuée : boîte au lettre inaccessible.

Dialogue SMTP

- Les 3 phases classiques de dialogue se retrouvent au niveau SMTP :
  - Etablissement de la connexion au niveau SMTP et identification de la source et de la destination ;
  - Envoi du message avec les différents en-tête RFC 822 et RFC 1521 ;
  - Libération de la connexion.



Le protocole ESMTP

- *Extended Simple Mail Transport Protocol*, RFC 1425
- Accessible via la requête initiale EHLO
- ESMTP rajoute des extensions au protocole SMTP
  - Déclaration de la taille des messages : SMTP détruit les messages trop longs et ne peut prévenir le client.
  - Transport sur 8 bits : envoi de corps de messages contenant des caractères codés sur 8 bits, sans avoir à utiliser l'encodage "base64", "quoted-printable" ou un autre...
  - Codes d'erreur plus complets et plus précis.
  - Gestion des accusés de réception.

Le protocole POP3

- POP3 (*Post Office Protocol version 3*) a été conçu pour récupérer le courrier sur une machine distante pour un utilisateur non connecté en permanence à Internet. Il gère :
  - l'authentification, c'est-à-dire la vérification du nom et du mot de passe ;
  - la réception des courriers et fichiers attachés à partir du serveur de messagerie ;
  - la réception de messages d'erreur ou d'acquittement.
- L'envoi de messages n'est pas supporté par le protocole POP3.
- Il n'est pas sécurisé : les messages sont transférés en clair
- Il est nécessaire de télécharger l'intégralité du courrier sur la station avant la lecture, sans possibilité de manipuler directement les messages sur le serveur.

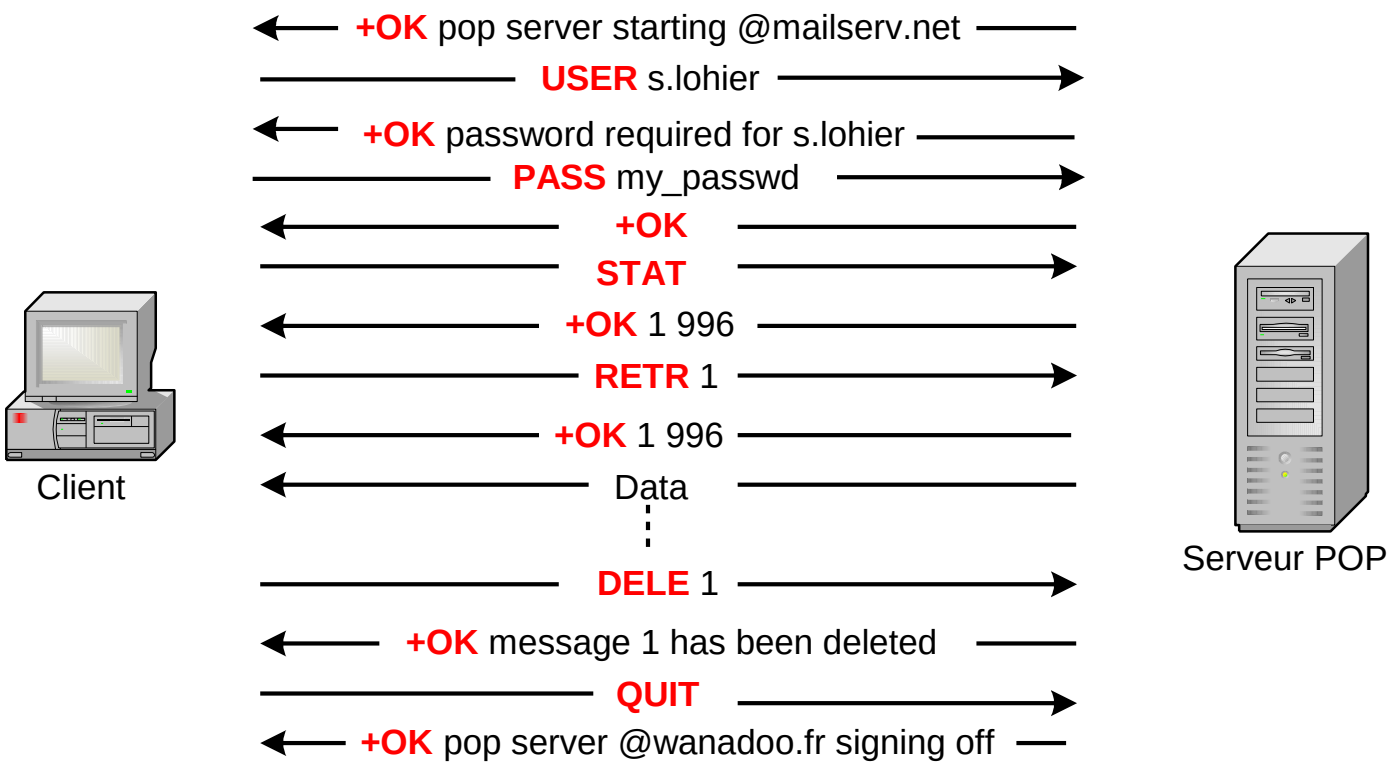
Commandes POP3

- Les commandes POP3 (RFC 1939) reprennent la syntaxe sur 4 lettres de SMTP.
- Les réponses du serveur sont transmises sous forme d'une chaîne de caractères sont de deux types : +OK et – ERR suivi d'un texte

Commande	Fonction
USER nom	Spécifie une boîte aux lettres.
PASS password	Spécifie un mot de passe.
STAT	Permet de récupérer le nombre et la taille des messages en attente.
LIST [msg]	Demande des informations sur le message dont le numéro est fourni.
RETR msg	Permet de récupérer une liste de messages.
DELE msg	Suppression du message spécifié.
QUIT	Le serveur supprime les messages lus et referme la connexion.

Dialogue POP3

- La première trame correspond à la réponse (+OK) à une demande de connexion TCP au serveur POP3.
- Le client POP3 s'identifie ensuite au serveur, ce dernier demande un mot de passe au client qui lui transmet sous forme non crypté sur le réseau.
- Après acceptation, le client peut relever le courrier choisi à l'aide de la commande RETR (sans la commande DELE le courrier reste sur le serveur).



Le protocole IMAP

- Le protocole IMAP (*Interactive Mail Access Protocol*) est un protocole destiné à la lecture interactive des messages.
- Le système webmail fonctionne avec le protocole IMAP.
- Parmi les principales fonctionnalités d'IMAP :
  - Lecture des objets des messages seulement (sans le corps)
  - création de dossiers sur le serveur
  - déplacement de messages sur le serveur d'un dossier à l'autre
  - effacement de message sans l'avoir lu
  - lecture des messages en les laissant sur le serveur
  - marquage des messages sur le serveur ...

Les commandes IMAP

- La structure des commandes IMAP est proche de celle des commandes POP3.

- Les commandes supplémentaires permettent d'effectuer des opérations telles que :
  - la création, la suppression ou le changement de nom de boîtes aux lettres ;
  - la vérification de la présence de nouveaux messages ;
  - la suppression définitive de messages ;
  - la pose et la suppression de marqueurs (flags) ;
  - la recherche de messages ;
  - la récupération de parties de messages.

Quelques commandes IMAP de base :

Authentification auprès du serveur :

Login <user> <mot_de_passe>

Choix d'une boite aux lettres :

Select inbox

La commande Fetch est très puissante elle permet de faire de nombreuses et diverses sélections auprès de la liste des messages et des messages eux-mêmes :

Fetch <id(s)_msg(s)> <action>

Quitter la session de dialogue avec le serveur :

Logout

Exemple de dialogue IMAP

```
* OK [CAPABILITY IMAP4REV1...
001 login testimap2 testimap2
001 OK [CAPABILITY IMAP4REV1 ...
002 select INBOX
* 1 EXISTS
* NO Trying to get mailbox lock from process 28032
* 0 RECENT
* OK [UIDVALIDITY 1068367935] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
002 OK [READ-WRITE] SELECT completed

003 fetch 1 BODY[HEADER]
* 1 FETCH (BODY[HEADER] {471)
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim
3.35 #1 (Debian))
        id 1AIlIA-0007Hi-00
        for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 09:51:30 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1AIlIA-0007Hi-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 09:51:30 +0100
)
003 OK FETCH completed

004 fetch 1 BODY[TEXT]
* 1 FETCH (BODY[TEXT] {8)
coucou
)
004 OK FETCH completed
005 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
005 OK LOGOUT completed
```

connexion au serveur

sélection d'un dossier

pour lire l'en-tête de l'unique message disponible

pour lire le texte du message

déconnexion du serveur

Exercice 3 : Architecture

Supposez qu'Alice (Alice@gmail.com) utilise le webmail de Gmail pour envoyer un message à Bob@o2switch.fr.

Celui-ci accède à son courrier électronique avec le client de messagerie courrier sur son PC.

Décrire le parcours du message entre les deux systèmes terminaux, en précisant bien les protocoles utilisés entre les machines.