Prise de contrôle à distance avec Linux

Version 2008-05-15

par Olivier ALLARD-JACQUIN
Mail : olivieraj@free.fr
Web : http://olivieraj.free.fr/

Le 9 avril 2008
http://www.guilde.asso.fr/

Licence

- Ce document est publié sous la Licence de Libre Diffusion de Documents (LLDD) :
  http://olivieraj.free.fr/share/html/license/lldd.html

Remerciements

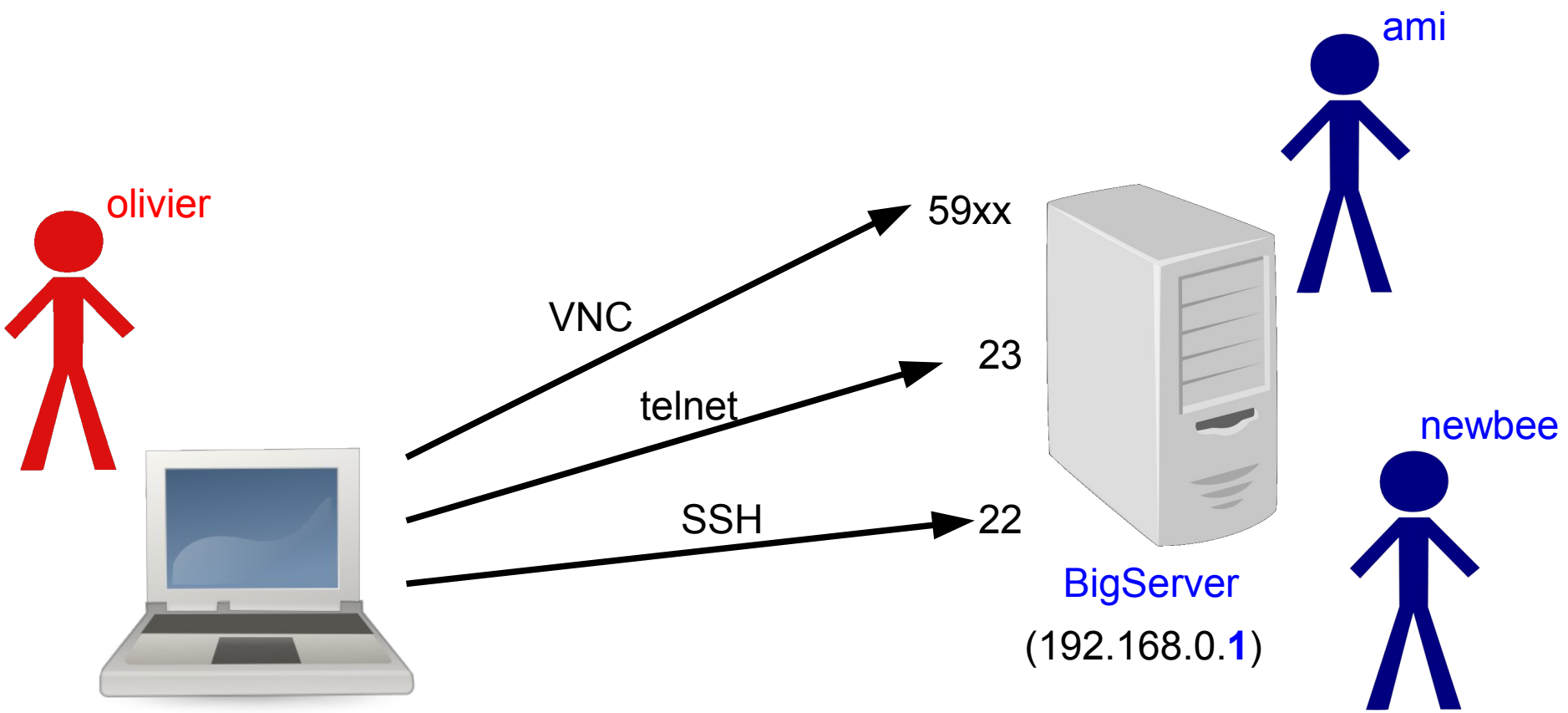
Plan

- Introduction
- Environnement de test
- L'ancêtre de la prise en main à distance : telnet
- La sécurité par le chiffrement : SSH
- Wrapper et SSH
- Échange de clés contre mot de passe
- SSH et tunnels
- X11Forwarding pour le lancement d'applications graphiques
- VNC et TightVNC : Le bureau à distance
  - VNC en mode serveur
  - Partage de bureau avec VNC
- VNC et SSH
- Autres solutions : NoMachine NX / FreeNX , RDP
- Prise en main à distance et firewall
- SSH callback, ou comment jouer les passes-murailles
- Autres utilisations du SSH callback
- Autres OS ?
- Conclusion

Introduction

- Pourquoi ?
  - Flexibilité
  - Coûts
  - Support & formation d'un utilisateur
- Comment ?
  - Mode client / serveur
  - Connexion IP
  - Droits d'accès
  - Protections des réseaux traversés

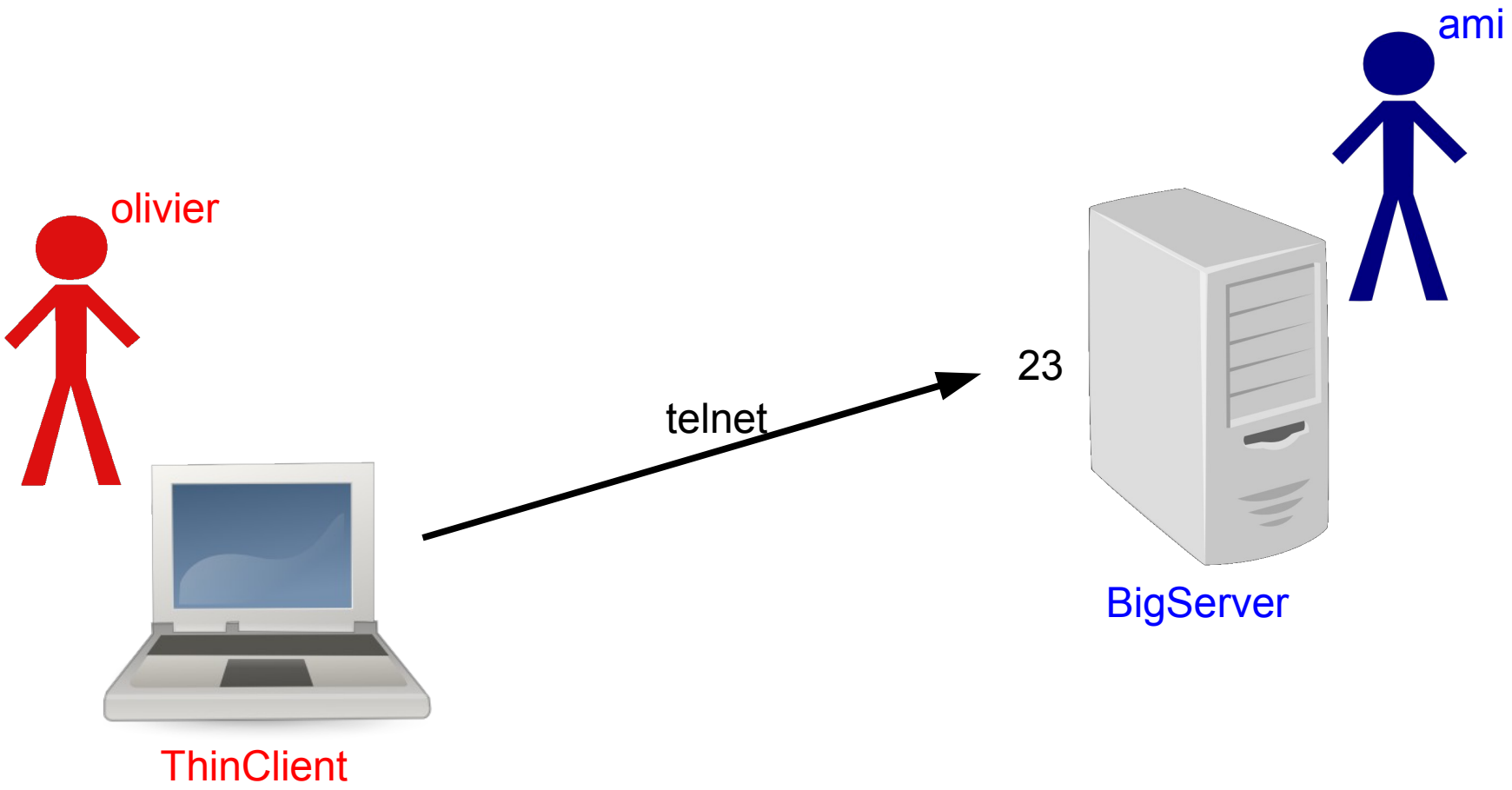
Environnement de test




```
root@ThinClient:~#
olivier@ThinClient:~$
```

```
root@BigServer:~#
ami@BigServer:~$
newbee@BigServer:~$
```


L'ancêtre de la prise en main à distance : telnet

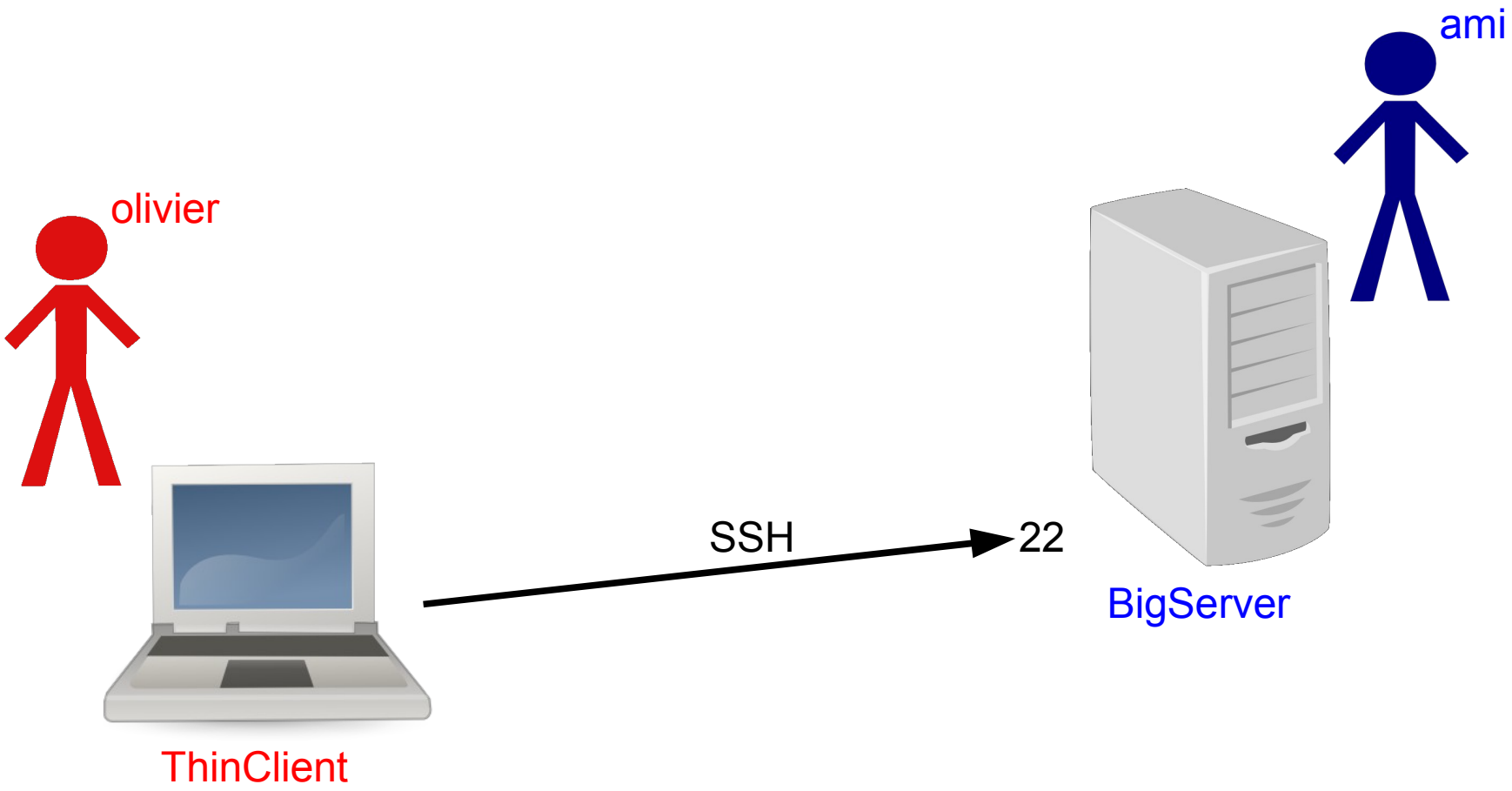



```
olivier@ThinClient:~$ telnet -l ami BigServer
```

L'ancêtre de la prise en main à distance : telnet

- Technique :
  - Communication : TCP/23
  - Serveur : `aptitude install telnetd`
  - Client : `aptitude install telnet`
- Les plus :
  - Simplicité du protocole et des clients/serveurs
  - Fiabilité
  - Usage multiple du client (pop, smtp ...)
- Les moins :
  - Confidentialité
  - Pas d'accès graphique

La sécurité par le chiffrement : SSH




```
olivier@ThinClient:~$ ssh ami@BigServer
```


La sécurité par le chiffrement : SSH

- Technique :
  - Communication : TCP/22
  - Serveur : `aptitude install openssh-server`
  - Client : `aptitude install openssh-client`
- Configuration du /etc/ssh/sshd_config :
  - Port 22
  - Protocole 2
  - PermitRootLogin no
  - X11Forwarding yes
  - AllowUsers xxxx yyyy zzzz
  - Subsystem sftp /usr/lib/openssh/sftp-server

La sécurité par le chiffrement : SSH

- Les plus :
  - Confidentialité (chiffrement RSA, DSA)
  - Authentification des machines et des utilisateurs (~/.ssh/know_hosts)
  - Echange de clés X, tunnels, ... (mais n'anticipons pas !)
- Les moins :
  - Puissance de calcul nécessaire au chiffrement
  - Complexité

Wrapper et SSH

- Inetd et Xinetd pour l'emballage de protocoles
- Les plus :
  - Sécurité accrue
  - Filtrage suivant des paramètres
  - Log
  - Libération de charge mémoire
  - Modification des paramètres serveurs
- Les moins :
  - Pas adapté à tous les protocoles (portmap, HTTP(S)...)

Wrapper et SSH

- Exemple : /etc/xinetd.d/sshd

```
service ssh
{
 id                  = sshd
 disable             = no
 type                = UNLISTED
 port                = 22
 socket_type         = stream
 wait                = no
 user                = root
 server              = /usr/sbin/sshd
 server_args         = -i
 log_on_success      += DURATION USERID
 log_on_failure      += USERID
 nice                = 10
}
```

Avant

```
# netstat -taupn|grep :22
tcp   0   0 :::22   :::*   LISTEN   11451/sshd
```

Après

```
# netstat -taupn|grep :22
tcp   0   0 :::22   :::*   LISTEN   25624/xinetd
```

Pendant

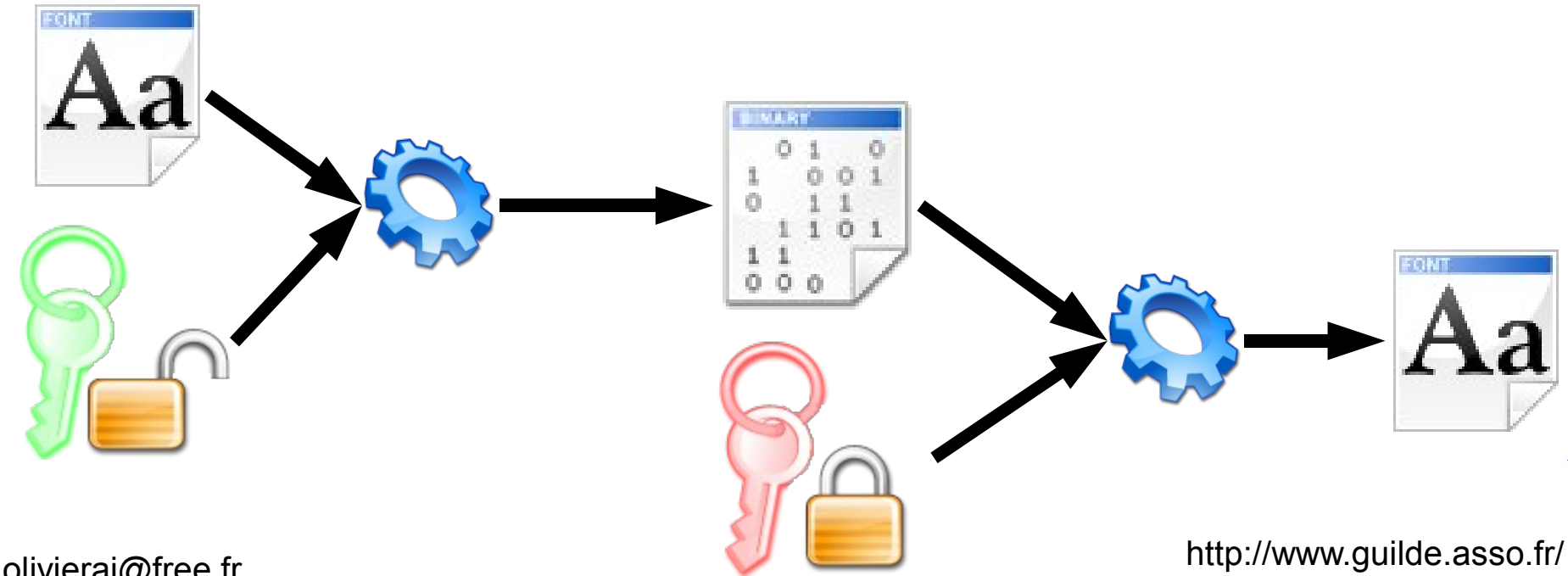
```
# netstat -taupn|grep :22
tcp   0   0   0.0.0.0:22   0.0.0.0:*   LISTEN   32220/xinetd
 tcp   0   0   x.x.x.x:22   x.x.x.y:46204   ESTABLISHED 32240/sshd: ami
```

Échange de clés contre mot de passe

- SSH garantie déjà la confidentialité
- Avec les clés, SSH rajoute l'authenticité des utilisateurs
- Le secret : Une paire de clés, publique/privée
- DSA, RSA
- Connexion sans échange de mot de passe (~/.ssh/authorized_keys)

Échange de clés contre mot de passe

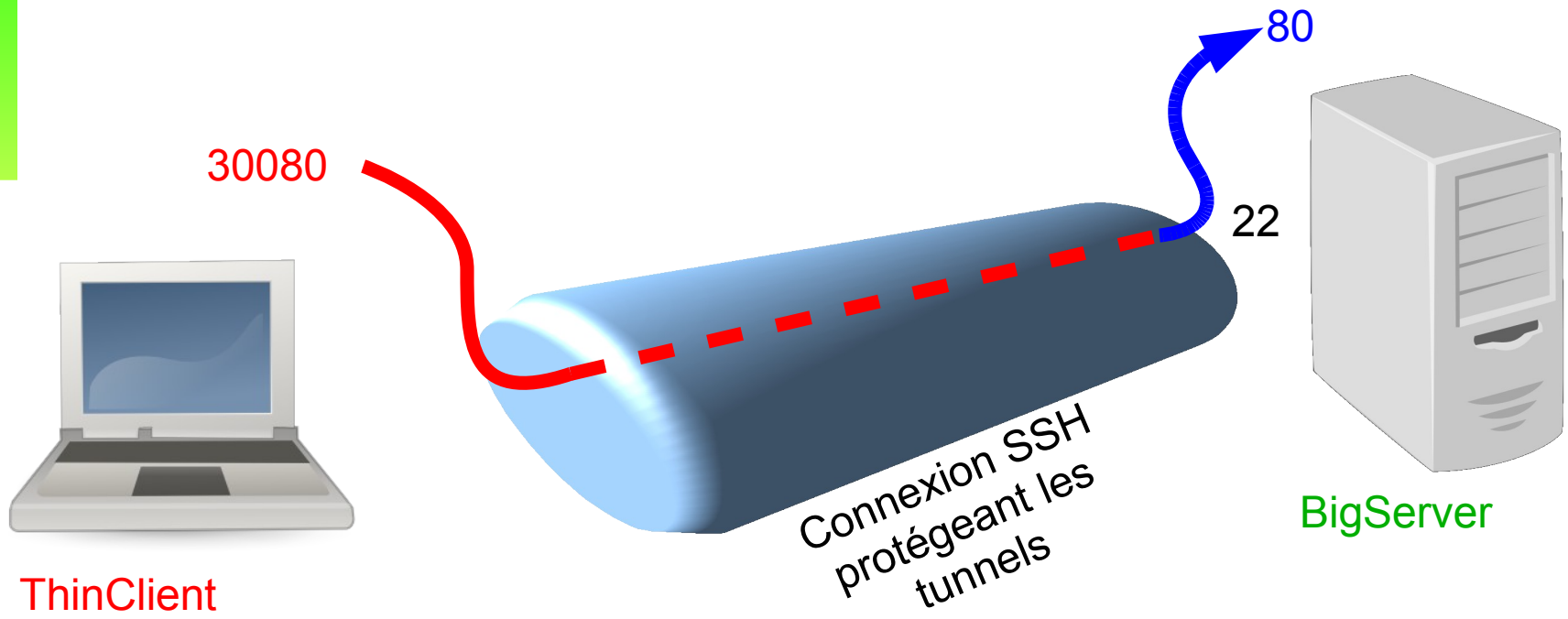
- Clé publique: A distribuer librement 
- Clé secrète: 
  - A conserver jalousement
  - Optionnel: "Phrase secrète" sur la clé privée



SSH et tunnels

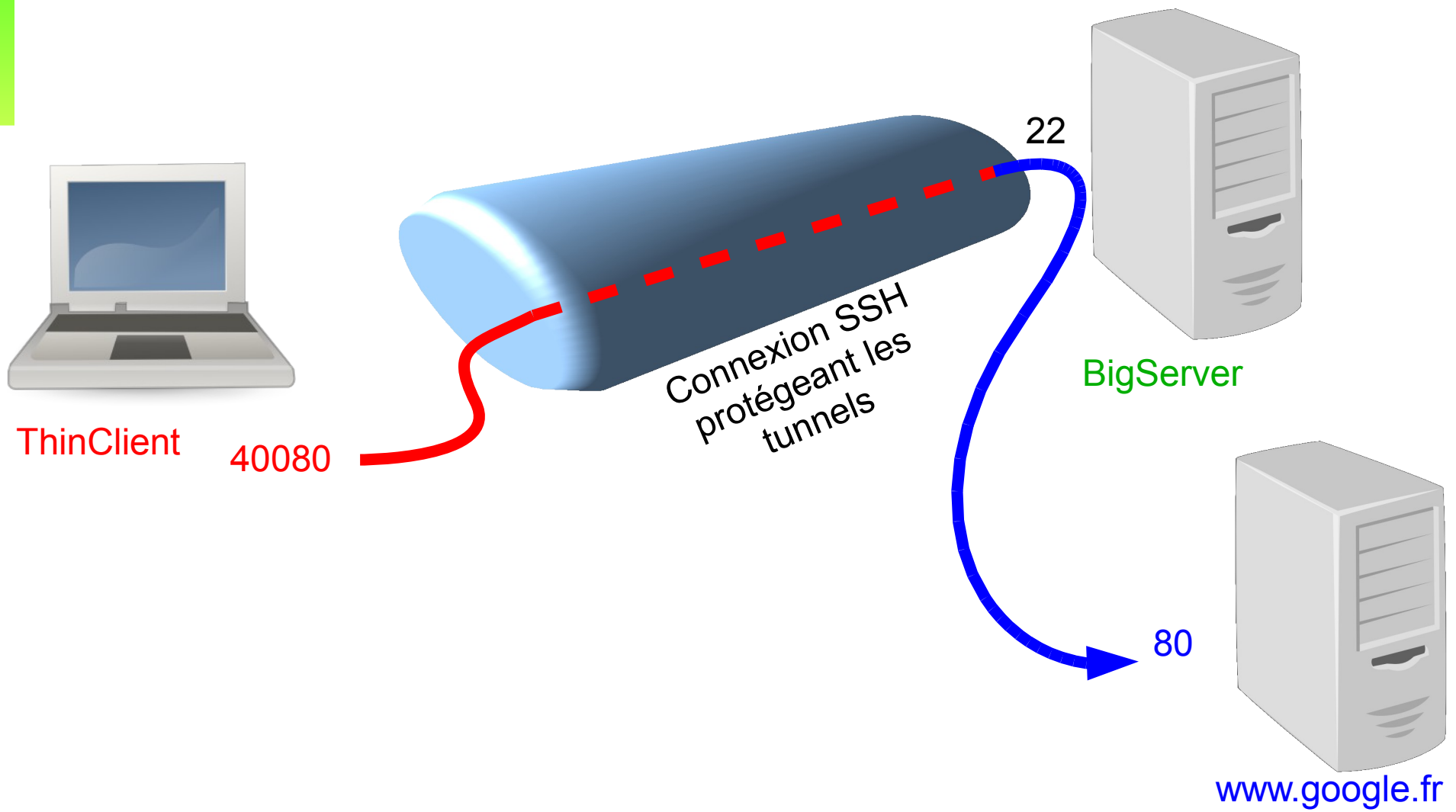
- But : Rediriger des flux TCP à travers une connexion SSH
- 2 types de tunnels :
  - Redirection locale (-L)
  - Redirection à distance (-R : "Remote")

SSH et tunnels : Redirection locale



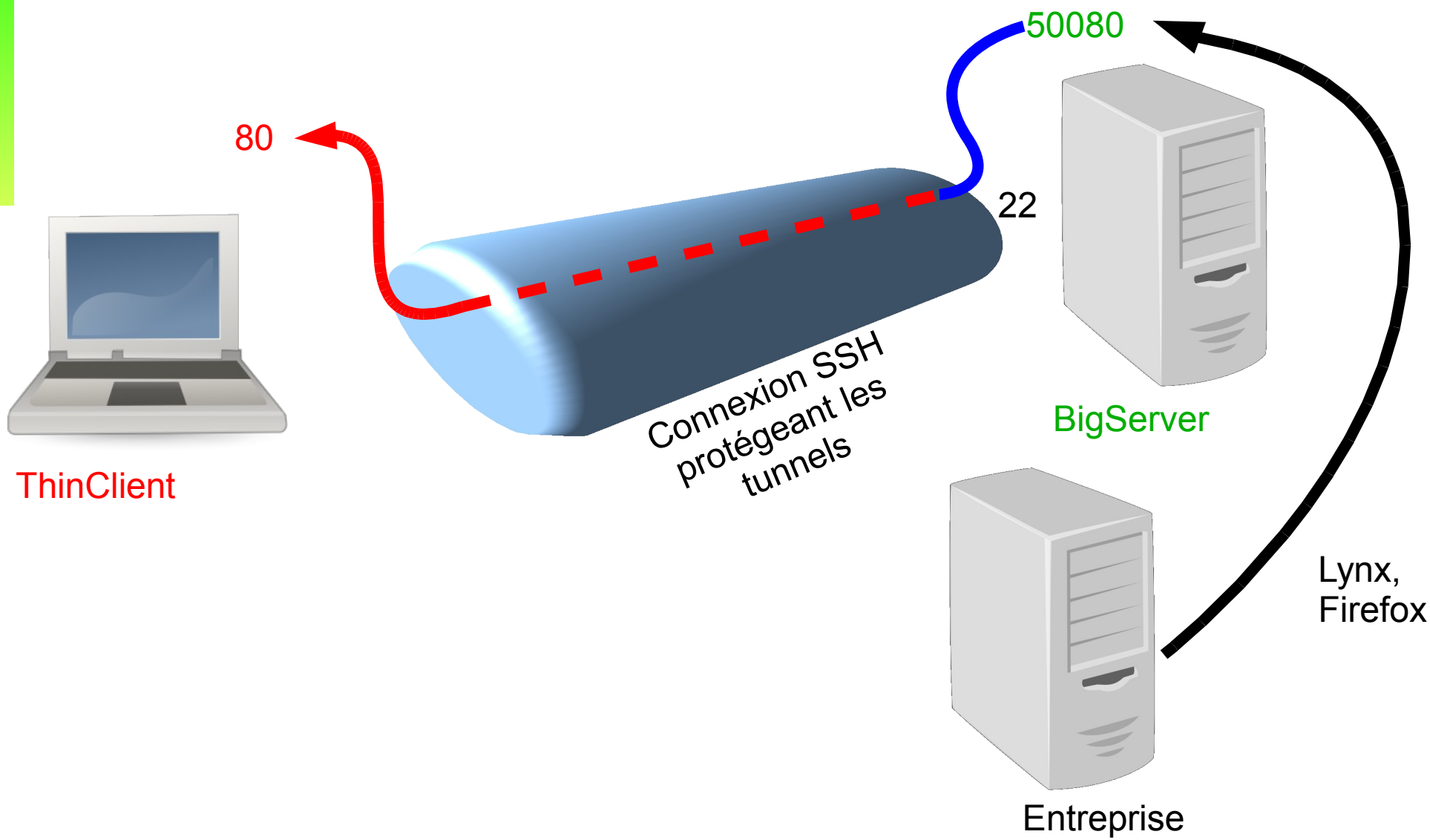

```
olivier@ThinClient:~$ ssh -L 30080:localhost:80
                          ami@BigServer
```


SSH et tunnels : Redirection locale




```
olivier@ThinClient:~$ ssh -L 40080:www.google.fr:80
                          ami@BigServer
```

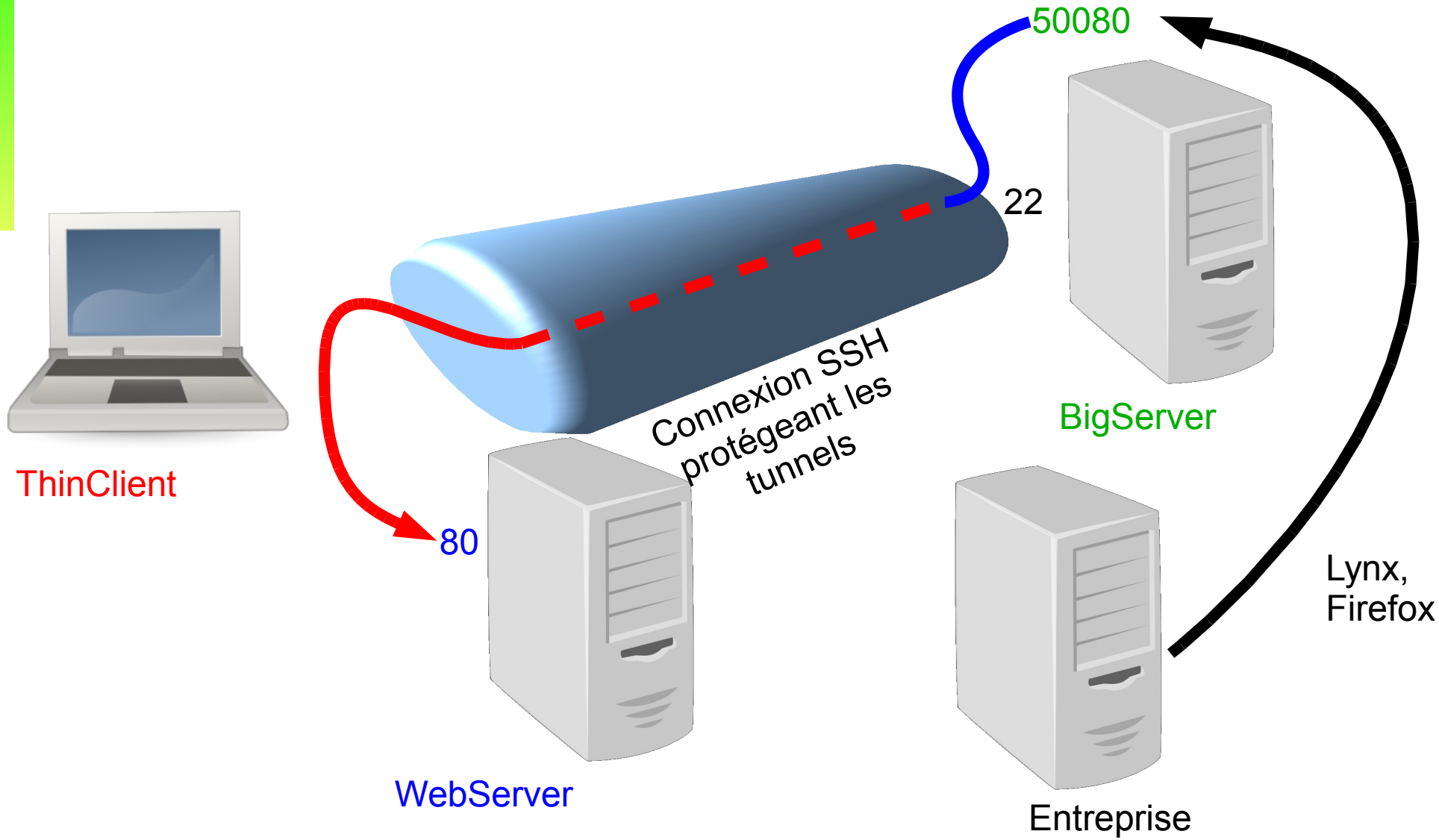

SSH et tunnels : Redirection à distance




```
olivier@ThinClient:~$ ssh -R 50080:localhost:80
                      ami@BigServer
```

olivieraj@free.fr

SSH et tunnels : Redirection à distance




```
olivier@ThinClient:~$ ssh -R 50080:WebServer:80
                      ami@BigServer
```

olivieraj@free.fr

X11Forwarding pour le lancement d'applications graphiques

-- Rappels X11 / Xorg --

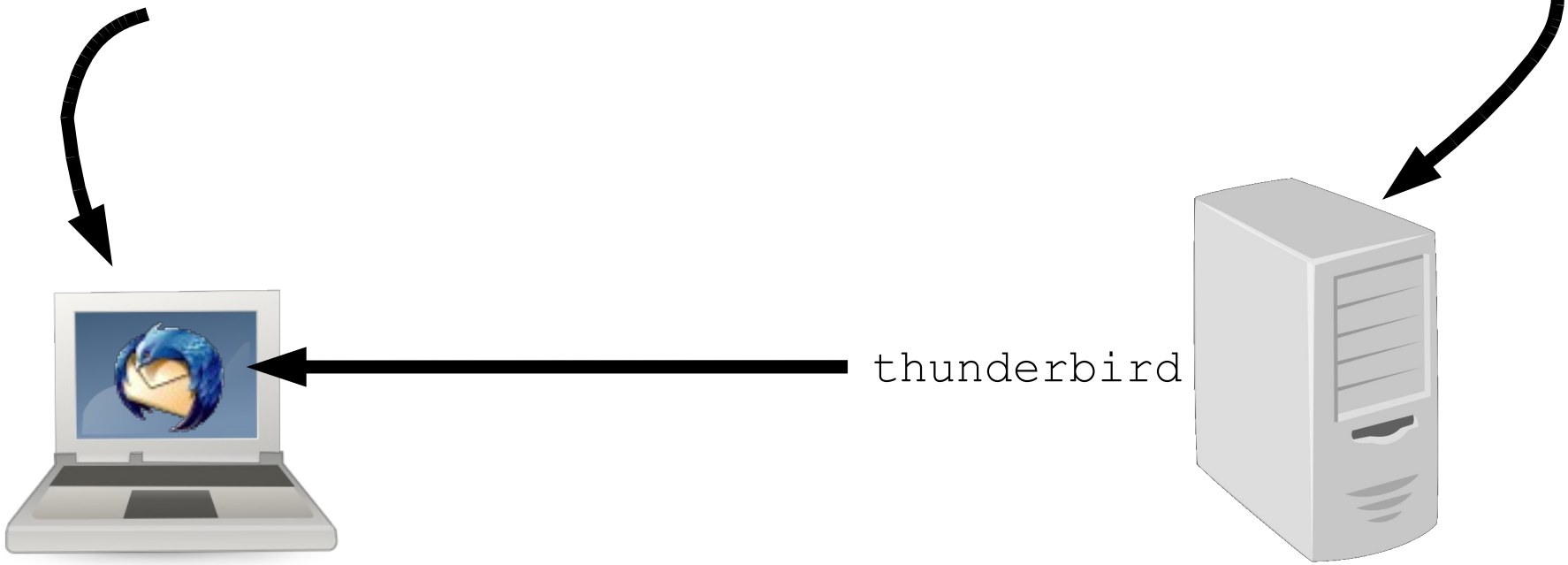
- X11 / Xorg : Protocole conçu sur un modèle client / serveur
- Mode de fonctionnement :
  - Local : Utilisation de sockets
  - À distance : TCP/6000
- Modèle spécifique :
  - Applications graphiques : Clients X
  - X11 (transcription des commandes X11 en instructions matérielles) : Serveur X

X11Forwarding pour le lancement d'applications graphiques

-- Export display --

- Application graphique lancée sur la machine à distance (client X) `export DISPLAY=ThinClient:0.0`
- Affichage sur la machine locale (serveur X)

`xhost +BigServer`

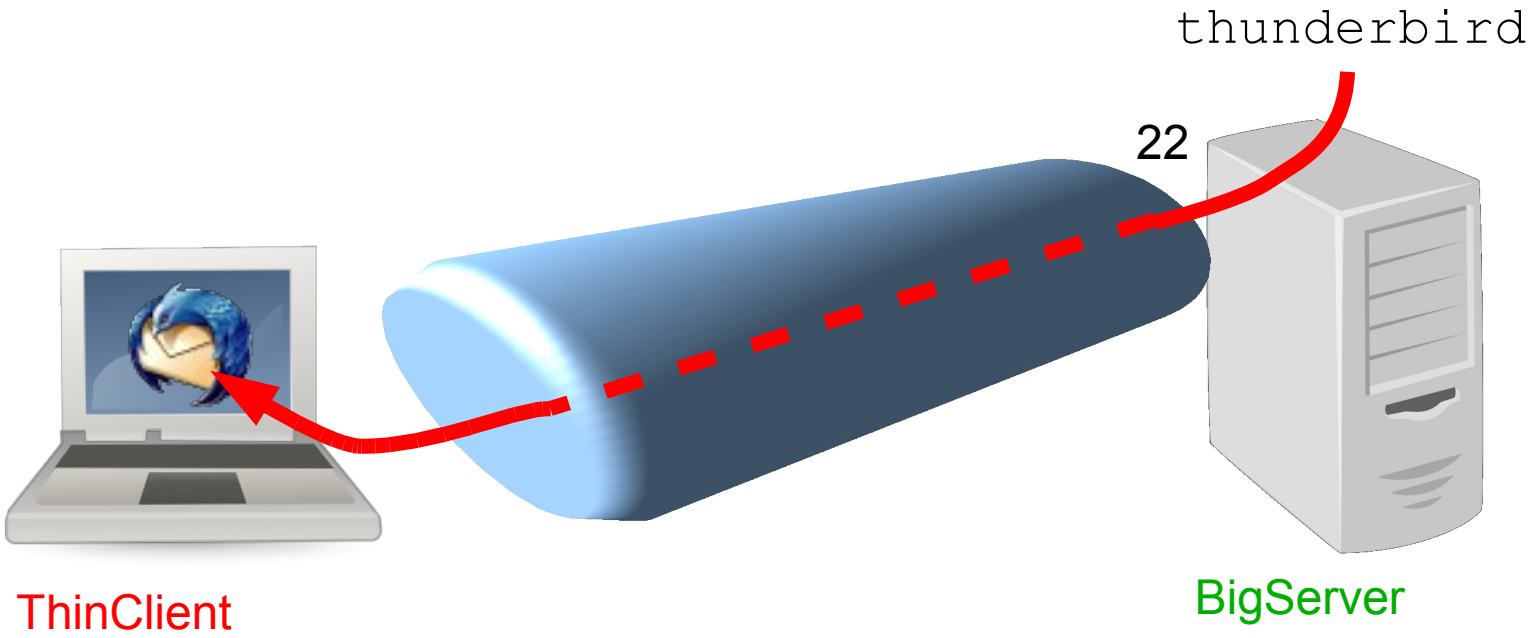


olivieraj@free.fr

X11Forwarding pour le lancement d'applications graphiques

-- SSH et X11Forwarding --

- /etc/ssh/sshd_config : X11Forwarding yes
- ssh -X ami@BigServer



olivieraj@free.fr

X11Forwarding pour le lancement d'applications graphiques

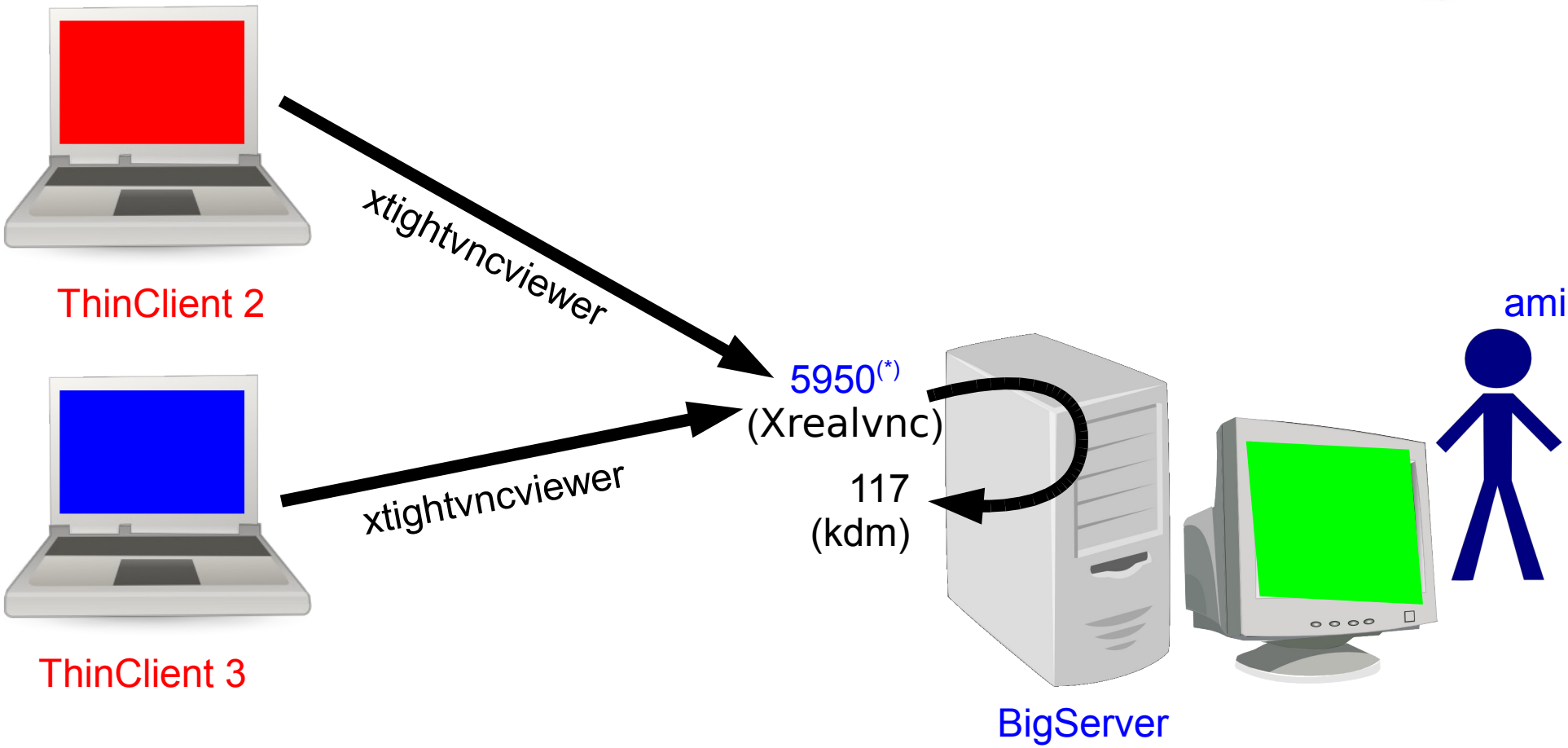
-- SSH et X11Forwarding --

- Les plus :
  - Simplicité à mettre en oeuvre
  - Simplicité à l'utilisation
- Les moins :
  - Bande passante utilisée. Réservé à un usage local
  - Cela reste du protocole X

VNC en mode serveur

- VNC : Protocole multi-plateforme d'affichage à distance (Linux, Windows, MacOS...)
- Optimisation de trafic avec TightVNC (xtightvncviewer)
- Mode serveur : Xrealvnc
- Dans ce mode, VNC est utilisé comme un pont entre le protocole XDMCP et le client VNC

VNC en mode serveur




```
ami@BigServer:~$ x11vnc -display :0 -loop(**)
olivier@ThinClient2|3:~$ xtightvncviewer BigServer:50(*)
```

(*): La relation entre le numéro du port TCP du serveur VNC et le "numéro d'écran" VNC se calcul comme suit: Numéro_écran_VNC = Port_TCP_serveur_VNC - 5900.

(**): Le "-loop" permet de redémarrer automatiquement le serveur x11vnc lorsque la connexion avec le client VNC s'interrompt.

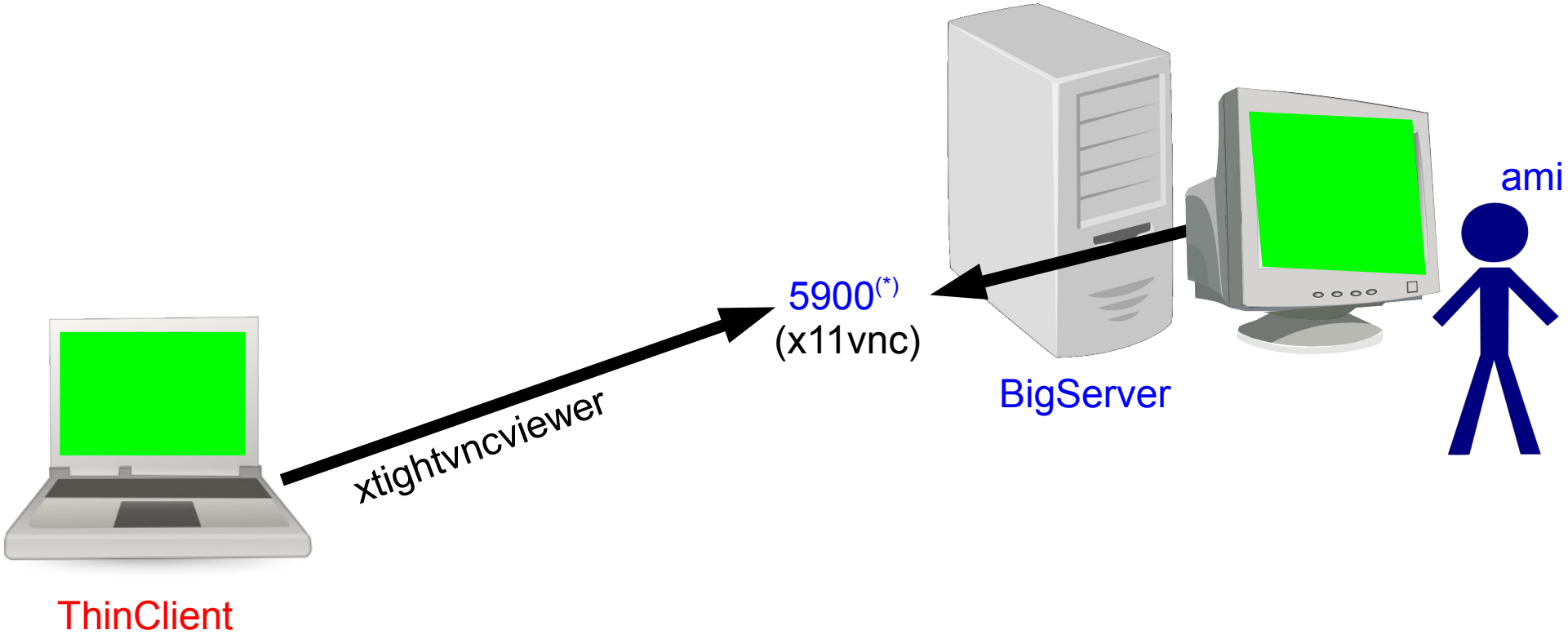
VNC en mode serveur

- Les plus :
  - Multi-utilisateurs
  - Mutli-résolutions
- Les moins :
  - Aucun partage de l'écran local
  - Inconsistance des sessions
  - Complexité à mettre en œuvre :
    - /etc/xinetd/vnc
    - Gestionnaire de session (gdm, kdm, xdm...) en mode XDMCP

Partage de bureau avec VNC

- Partage de bureau : 2 utilisateurs pour la même session
- X11vnc
- Lancement par le possesseur de l'écran local

Partage de bureau avec VNC




```
ami@BigServer:~$ x11vnc -display :0 -loop(**)

olivier@ThinClient:~$ xtightvncviewer BigServer:0(*)
```

(*): La relation entre le numéro du port TCP du serveur VNC et le "numéro d'écran" VNC se calcul comme suit: Numéro_écran_VNC = Port_TCP_serveur_VNC - 5900.

(**): Le "-loop" permet de redémarrer automatiquement le serveur x11vnc lorsque la connexion avec le client VNC s'interrompt.

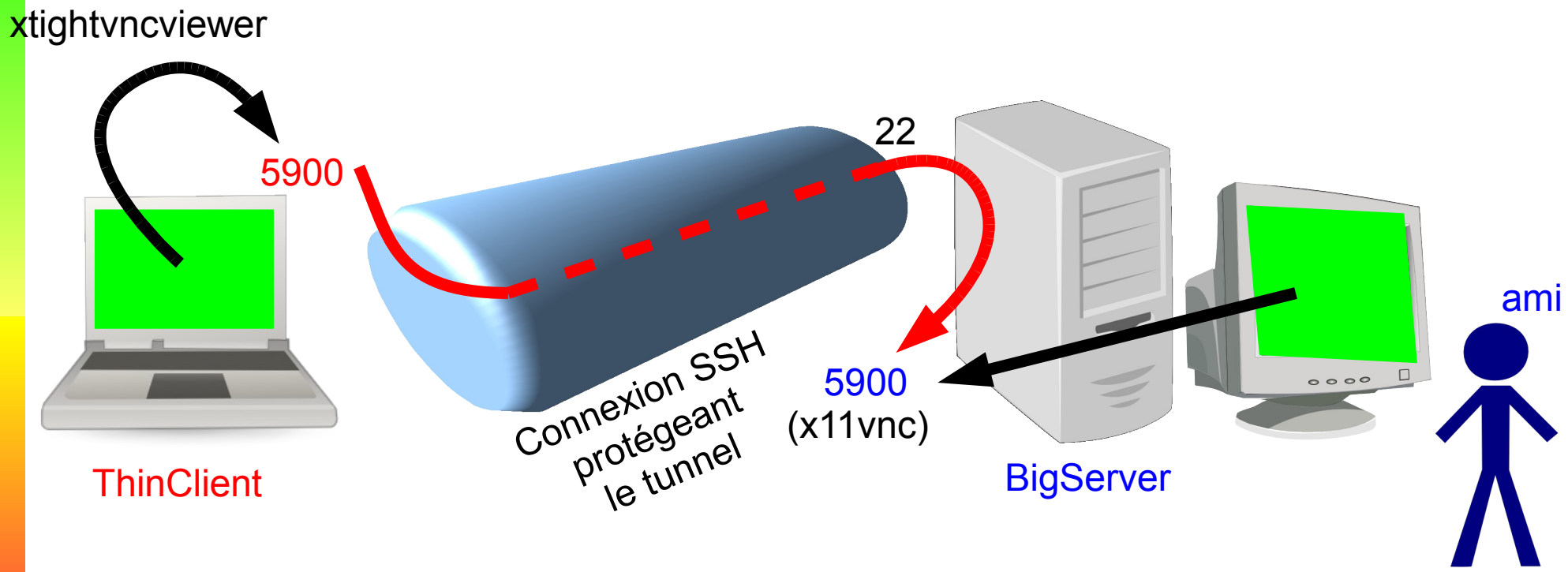
Partage de bureau avec VNC

- Les plus :
  - Partage de session
  - Consistance (partielle)
  - Simple à mettre en œuvre
- Les moins :
  - Résolution fixe
  - Absence de son

VNC et SSH

- Constat : VNC n'est pas chiffré
- Solution : Utiliser VNC à travers un tunnel SSH
- Les plus :
  - Sécurisation des transactions (chiffrement)
  - Sécurisation du serveur (SSH et non pas VNC)
- Les moins :
  - Puissance de calcul
  - Léger overhead

VNC et SSH




```
ami@BigServer:~$ x11vnc -display :0 -loop -allow localhost(*)

olivier@ThinClient:~$ ssh -L 5900:localhost:5900
                                    ami@BigServer
olivier@ThinClient:~$ xtightvncviewer localhost:0
```

(*): N'autorise les connexions au serveur VNC que sur l'interface loopback. Une personne située à distance ne pourra pas donc contacter directement le serveur VNC. Seule une connexion à travers un tunnel SSH fonctionnera.

Autres solutions : NoMachine NX / FreeNX

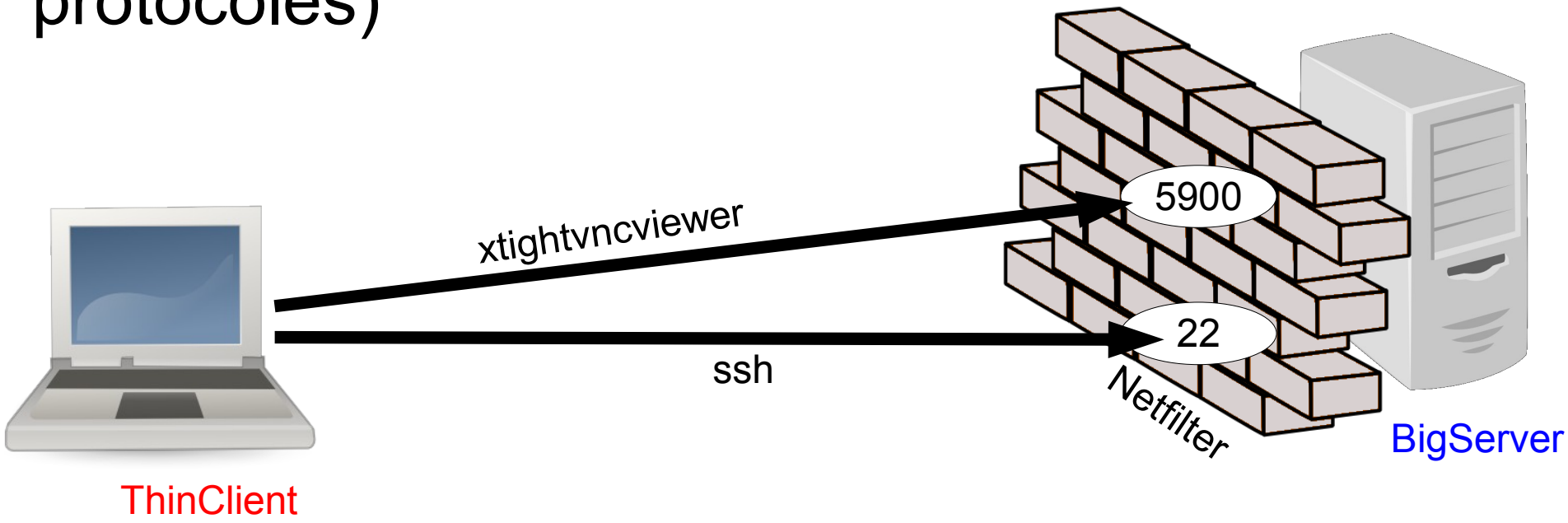
- NoMachine NX : Solution graphique et gratuite de prise en main à distance (http://www.nomachine.com/)
  - Bande passante optimisée, très réactif
  - Chiffrement des connexions
  - Gestion du son, partage de fichier
  - Reprise de session
  - Evolution rapide
  - ...
- FreeNX, implémentation libre de NoMachine

Autres solutions : RDP

- RDP: Remote DesktoP
- Protocole Microsoft® de prise en main à distance
- Pas de chiffrement
- 2 versions :
  - Multi-session : Windows XP Pro & 2003 serveur
  - Partage de bureau : Windows XP Familiale
- Clients Linux : rdesktop, grdesktop (Gnome), kdrc (KDE), NX...

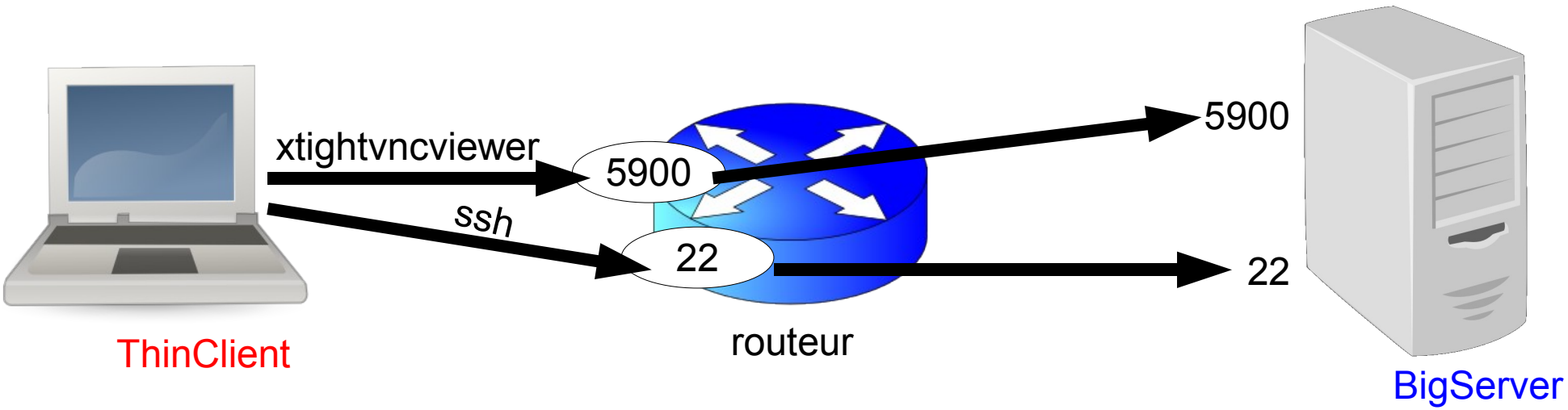
Prise en main à distance et firewall

- Ouverture de port(s) pour le firewall (Netfilter)
- Filtrage éventuel sur le parcours (ports, protocoles)



Prise en main à distance et firewall

- Port forwarding pour les routeurs domestiques, ou les passerelles Linux



SSH callback, ou comment jouer les passes-murailles

- Support utilisateurs (*), les difficultés :
  - Mode console insuffisant (formation à distance, applications graphiques...)
  - Connexion à la demande de l'utilisateur (*)
  - Firewall / routeurs personnels (ex : "box" ADSL)
  - Confidentialité, authentification
  - Failles de sécurité

(*): Par la suite, l'utilisateur sera appelé "newbee"

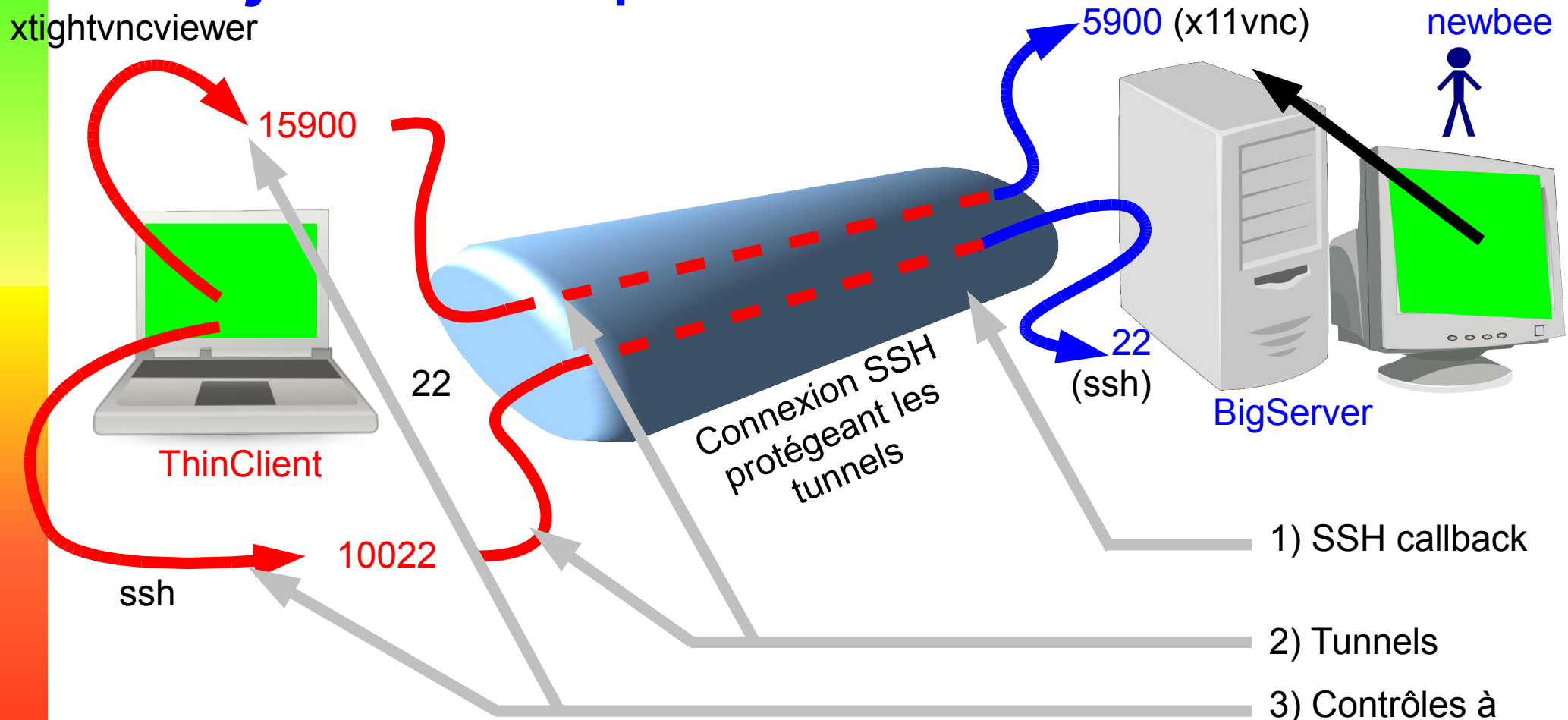
SSH callback, ou comment jouer les passes-murailles

- Solution : SSH callback(*)
- Newbee
  - Démarrage x11vnc
  - Connexion SSH à destination de l'administrateur
  - Établissement des tunnels
- Administrateur
  - "Remonte" les tunnels SSH
  - Connexion SSH et / ou VNC

(*): Callback signifie "rappel"

olivieraj@free.fr

SSH callback, ou comment jouer les passes-murailles




```
newbee@BigServer:~$ ssh -R 15900:localhost:5900
                        -R 10022:localhost:22
                        newbee@ThinClient
```

```
olivier@ThinClient:~$ xtightvncviewer localhost:10000
olivier@ThinClient:~$ ssh -oPort=10022 newbee@localhost
```

SSH callback, ou comment jouer les passes-murailles

-- Pseudo-shell --

```
#include <stdio.h>
#include <unistd.h>

int main (int argc, char* argv[]) {
  printf("Le tunnel SSH callback est actif\n\
L'utilisateur à distance peut maintenant se connecter\n\
Vous pouvez arrêter cette connexion en tapant plusieurs fois Ctrl+C\n");
  sleep(-1);
  return 0;
}
```

```
root@ThinClient:~# gcc -Wall ssh-cb.c
                        -o /usr/local/sbin/ssh-cb
```

SSH callback, ou comment jouer les passes-murailles

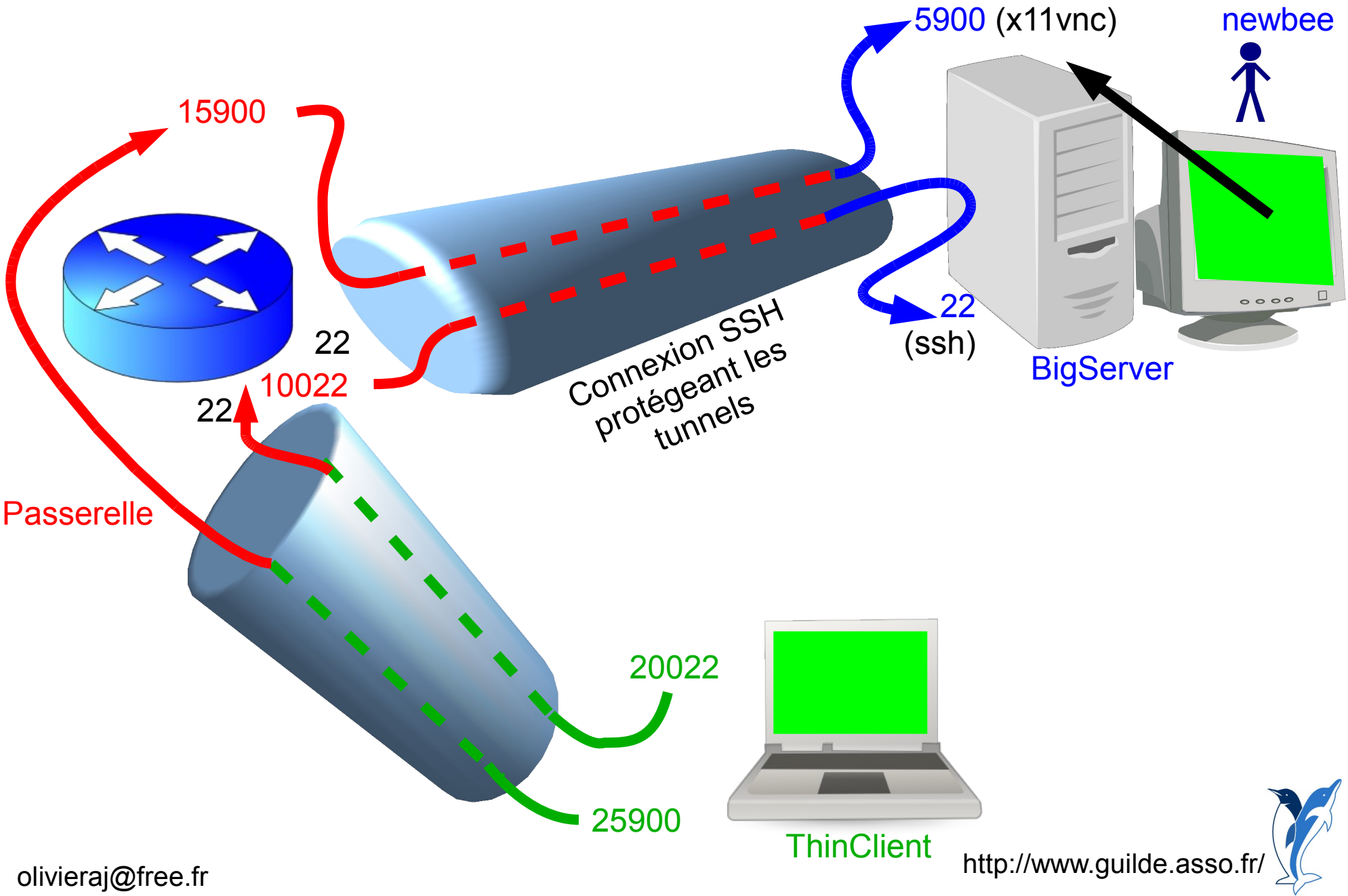
-- Script "un clic" pour "newbee" --

```
#!/bin/bash -norc

SERVER_NAME=ThinClient
SERVER_PORT=22
ACCOUNT_NAME=newbee

REMOTE_SSH_PORT=10022
REMOTE_VNC_PORT=15900

echo + Starting x11vnc server...
x11vnc -rfbauth ~/.vnc/passwd -display :0 -loop -allow localhost &
sleep 5s

echo
echo + Calling remote callback host...
ssh -p $SERVER_PORT $ACCOUNT_NAME@$SERVER_NAME \
        -R $REMOTE_SSH_PORT:localhost:22 \
        -R $REMOTE_VNC_PORT:localhost:5900

echo + Kill x11vnc server...
killall x11vnc || (sleep 2s ; killall -9 x11vnc)
```

SSH callback, ou comment jouer les passes-murailles

- Les plus :
  - Aucun port ouvert coté "newbee"
  - Contournement des firewall / routeur
  - Chiffrement
  - Mode console et graphique
- Les moins :
  - Complexité
  - Serveur SSH coté administrateur
  - Chroot des comptes "newbee"
  - Shell sécurisé pour les comptes "newbee"
  - Confiance mutuelle (ouverture de tunnels)

Autres utilisations du SSH callback



olivieraj@free.fr

Autres OS ?

- Besoins :
  - Client / serveur SSH
  - Client / serveur graphique
- Solutions :
  - Cygwin (http://www.cygwin.com/)
  - VNC
  - NoMachine / DRP
  - Tunnel SSH + X11 :

    http://resel.enst-bretagne.fr/configuration/xming/
  - ...

Conclusion

Bibliographie

- http://www.openssh.com/
- http://www.realvnc.com/
- http://www.tightvnc.com/
- http://www.nomachine.com/
- http://www.cygwin.com/
- http://olivieraj.free.fr/

Questions

olivieraj@free.fr